

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号
特開2001-189896
(P2001-189896A)

(43)公開日 平成13年7月10日(2001.7.10)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード*(参考)
H 0 4 N	5/445	H 0 4 N 5/445	Z
H 0 4 H	1/00	H 0 4 H 1/00	C
H 0 4 N	5/44	H 0 4 N 5/44	H
	7/025		A
	7/03		

審査請求 未請求 請求項の数69 O L 外国語出願 (全 130 頁) 最終頁に続く

(21)出願番号 特願2000-299286(P2000-299286)

(22)出願日 平成12年9月29日(2000.9.29)

(31)優先権主張番号 P Q 3 2 1 7

(32)優先日 平成11年9月30日(1999.9.30)

(33)優先権主張国 オーストラリア (A U)

(31)優先権主張番号 P Q 8 5 2 9

(32)優先日 平成12年7月3日(2000.7.3)

(33)優先権主張国 オーストラリア (A U)

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 ドン メイ ツァン

オーストラリア国 2113 ニュー サウス

ウェールズ州, ノース ライド, トーマ

ス ホルト ドライブ 1 キヤノン イ

ンフォメーション システムズ リサーチ

オーストラリア プロプライエタリー

リミテッド 内

(74)代理人 100076428

弁理士 大塚 康德 (外2名)

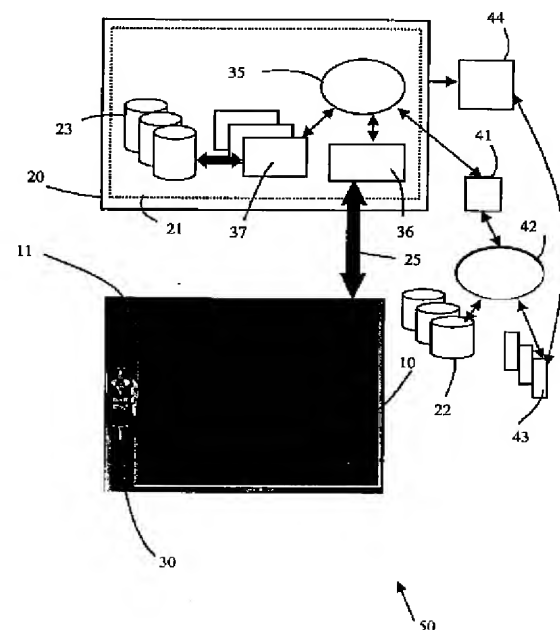
最終頁に続く

(54)【発明の名称】 番組選択支援システム

(57)【要約】 (修正有)

【課題】 多数の視聴可能な番組から適切な番組を視聴者に自動的に提案するテレビ・システムを提供する。

【解決手段】 システム50はDTVエージェント・システム21及び少なくとも1つの電子番組ガイド・データベース22を含む。アバタエージェント37内の学習モジュールはユーザによって見られた各番組に関連付けられた番組特性を記録し、これらの番組特性の組を形成する。各組の発生の頻度も決定される。推薦モジュールは、いくつかのタスクを使用して視聴者推薦のリストをコンパイルする。種々のタスクが定義され、各タスクは視聴者プロフィールを順序付ける方法の唯一の組み合わせと順序付けられた視聴者プロフィールをフィルタリングするための特定の適切性フィルタを定義する。番組推薦のリストを要求するユーザの入力で、タスクによって選択された組に最も適合する番組特性を持つ番組を見つけるためにEPGデータの検索が実行される。



【特許請求の範囲】

【請求項1】 番組のタイトル情報及び番組特性を少なくとも1つの番組ガイド・リストを含むE P G（電子番組ガイド）データとして利用可能なテレビ・システムで見る番組の選択を可能にする方法であって、前記テレビ・システムにおいてユーザが見た各番組と関連付けて複数の番組特性を記録するステップと、各組が前記番組特性の少なくとも2つからなる番組特性の組を形成するステップと、少なくとも各組を、特定の番組を見たいユーザの希望を表す順序付けられた値と関連付けるステップとを有し、番組推薦を要求するユーザの入力で、前記番組特性の組に最も適合する番組特性を有する番組を見つけるためにE P Gデータの検索を行い、前記番組特性の組に最も適合する番組の利用可能性を番組推薦として前記ユーザに通知することを特徴とする方法。

【請求項2】 前記検索は、ユーザの手動の操作に応じて決定されたユーザ情報及び前記番組特性の組に最も適合する番組を見つけるために行われることを特徴とする請求項1に記載の方法。

【請求項3】 前記ユーザ情報は、前記ユーザによって経験されるムードを含むことを特徴とする請求項2に記載の方法。

【請求項4】 前記番組推薦は、前記番組特性を最も多く有する前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項1に記載の方法。

【請求項5】 前記番組推薦は、前記順序付けられた値として最も低い値に関連付けられた前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項1に記載の方法。

【請求項6】 前記番組推薦は、前記順序付けられた値として最も高い値に関連付けられ、前記番組特性を最も多く有する、前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項1に記載の方法。

【請求項7】 前記番組推薦は、前記順序付けられた値として最も低い値に関連付けられ、前記番組特性を最も多く有する、前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項1に記載の方法。

【請求項8】 前記番組特性の組は、前記ユーザが見た少なくとも2つの番組が同じ番組特性を有することに応じて形成されることを特徴とする請求項1に記載の方法。

【請求項9】 番組のタイトル情報及び番組特性を少なくとも1つの番組ガイド・リストが含まれるE P G（電子番組ガイド）データとして入力可能なテレビ・システムで見る番組の選択を可能にする推薦システムであって、前記テレビ・システムにおいてユーザが見た各番組と関連付けて複数の番組特性を記録するメモリ手段と、各組が前記番組特性の少なくとも2つを有する番組特性

の組を形成し、少なくとも各組を、特定の番組を見たいユーザの希望を表す順序付けられた値と関連付ける処理手段と、

前記番組特性の組に最も適合する番組特性を有する番組を見つけるためにE P Gデータの検索を行う検索手段と、

番組推薦を要求するユーザの入力で、前記番組特性の組に最も適合する番組の有用性を番組推薦として前記ユーザに通知する画面上表示手段とを有することを特徴とする推薦システム。

【請求項10】 前記検索は、ユーザの手動の操作に応じて決定されたユーザ情報及び前記番組特性の組に最も適合する番組を見つけるために行われることを特徴とする請求項9に記載の推薦システム。

【請求項11】 前記ユーザ情報は、前記ユーザによって経験されるムードを含むことを特徴とする請求項10に記載の推薦システム。

【請求項12】 前記番組推薦は、前記番組特性を最も多く有する前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項9に記載の推薦システム。

【請求項13】 前記番組推薦は、前記順序付けられた値として最も低い値に関連付けられた前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項9に記載の推薦システム。

【請求項14】 前記番組推薦は、前記順序付けられた値として最も高い値に関連付けられ、前記番組特性を最も多く有する、前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項9に記載の推薦システム。

【請求項15】 前記番組推薦は、前記順序付けられた値として最も低い値に関連付けられ、前記番組特性を最も多く有する、前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項9に記載の推薦システム。

【請求項16】 前記番組特性の組は、前記ユーザが見た少なくとも2つの番組が同じ番組特性を有することに応じて形成されることを特徴とする請求項9に記載の推薦システム。

【請求項17】 番組のタイトル情報及び番組特性が少なくとも1つの番組ガイド・リストを含むE P G（電子番組ガイド）データとして利用可能なテレビ・システムで見る番組の選択を可能にするためのコンピュータ・プログラムが記録されたコンピュータ読み取り可能な媒体を有するコンピュータ・プログラム製品であって、前記テレビ・システムにおいてユーザが見た各番組と関連付けて複数の番組特性を記録するコンピュータ・プログラムコードと、各組が前記番組特性の少なくとも2つからなる番組特性の組を形成するコンピュータ・プログラムコードと、少なくとも各組を、特定の番組を見たいユーザの希望を

表す順序付けられた値と関連付けるコンピュータ・プログラムコードとを有し、

番組推薦を要求するユーザの入力で、前記番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行い、前記番組特性の組に最も適合する番組の有用性を番組推薦として前記ユーザに通知することを特徴とするコンピュータ・プログラム製品。

【請求項18】 前記検索は、ユーザの手動の操作に応じて決定されたユーザ情報及び前記番組特性の組に最も適合する番組を見つけるために行われることを特徴とする請求項17に記載のコンピュータ・プログラム製品。

【請求項19】 前記ユーザ情報は、前記ユーザによって経験されるムードを含むことを特徴とする請求項18に記載のコンピュータ・プログラム製品。

【請求項20】 前記番組推薦は、前記番組特性を最も多く有する前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項17に記載のコンピュータ・プログラム製品。

【請求項21】 前記番組推薦は、前記順序付けられた値として最も低い値に関連付けられた前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項17に記載のコンピュータ・プログラム製品。

【請求項22】 前記番組推薦は、前記順序付けられた値として最も高い値に関連付けられ、前記番組特性を最も多く有する、前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項17に記載のコンピュータ・プログラム製品。

【請求項23】 前記番組推薦は、前記順序付けられた値として最も低い値に関連付けられ、前記番組特性を最も多く有する、前記番組特性の組に最も適合する番組に基づくことを特徴とする請求項17に記載のコンピュータ・プログラム製品。

【請求項24】 前記番組特性の組は、前記ユーザが見た少なくとも2つの番組が同じ番組特性を有することに依拠して形成されることを特徴とする請求項17に記載のコンピュータ・プログラム製品。

【請求項25】 番組に関する情報を少なくとも1つの番組ガイド・リストを含むEPG（電子番組ガイド）として利用可能であり、ユーザが以前に見た各番組と関連付けられた複数の番組特性と、以前の番組要求に応じて行われた番組推薦に関して、複数の関係の各々と該関係に対応付けて前記ユーザの受諾の度合いとが記録されている、テレビ・システムで見る番組の推薦をユーザに行うための方法であって、

前記番組特性の組を形成するステップと、

前記関係から、前記受諾の度合いのうち最も高い度合いを有する少なくとも1つの関係を選択するステップと、前記選択された関係を使用して前記番組特性の組を順序付け、番組特性の順序付けられた組を生成するステップ

と、

前記選択された関係を使用して前記順序付けられた番組特性の組をフィルタリングし、番組特性のフィルタリングされた組を生成するステップと、

前記フィルタリングされた番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行い、最も高いフィルタリングされた番組特性の組を優先するステップと、

番組推薦を要求するユーザの入力で、前記フィルタリングされた組に最も適合する番組の有用性を番組推薦として前記ユーザに通知するステップとを有することを特徴とする方法。

【請求項26】 各組は、前記複数の番組特性を有することを特徴とする請求項25に記載の方法。

【請求項27】 前記受諾の度合いを更新する更なるステップを有することを特徴とする請求項25に記載の方法。

【請求項28】 前記関係の各々が少なくとも第1のタイプの関係と第2のタイプの関係の組合せを有し、前記第1のタイプの関係の前記度合いが、延長された期間にわたって行われたユーザの選択の結果として更新され、前記第2のタイプの関係の前記度合いが、より短い期間にわたって行われたユーザの選択の結果として更新されることを特徴とする請求項27に記載の方法。

【請求項29】 前記関係の各々は、受諾の最大の度合いを有する前記第1のタイプの関係と、全ての第2のタイプの関係の組合せを有することを特徴とする請求項28に記載の方法。

【請求項30】 前記関係は、各ユーザ独自に設定されることを特徴とする請求項25に記載の方法。

【請求項31】 前記番組特性の組の各々は、前記ユーザが以前に見た番組内における各組の発生頻度に関連付けられていることを特徴とする請求項25に記載の方法。

【請求項32】 前記番組特性の組の各々はそれに関連付けられた日付を有し、前記日付は前記ユーザが以前に見た番組内で各組が発生した時を示すことを特徴とする請求項25に記載の方法。

【請求項33】 前記関係の各々は、フィルタリングコマンド及び順序付けコマンドによって定義されていることを特徴とする請求項25又は26に記載の方法。

【請求項34】 前記フィルタリングコマンドは、時間スロット・フィルタ、重み係数フィルタ、人気フィルタ、緊急フィルタ、及びライフスタイル・フィルタを含むリストから選択されることを特徴とする請求項33に記載の方法。

【請求項35】 前記時間スロット・フィルタは、推薦が要求される日又は時間にふさわしい日又は時間に依存し、前記日又は時間は前記番組特性の組に含まれ、各組が前記ユーザが以前に見た番組内で発生した時を示し、

前記重み係数フィルタは前記ユーザが以前に見た番組内での各組の発生頻度に依存し、

前記人気フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組が人気のあるイベント／番組に対応することを示す人気値に依存し、

前記緊急フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組が緊急番組に対応することを示す緊急値に依存し、

前記ライフスタイル・フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組がライフスタイルのタイプに対応することを示すライフスタイル値に依存することを特徴とする請求項34に記載の方法。

【請求項36】 前記順序付けコマンドは、重み係数順序付け、新しさ順序付け、重み新しさ順序付けを含むリストから選択されることを特徴とする請求項33に記載の方法。

【請求項37】 前記重み係数順序付けは、前記ユーザが以前に見た番組内での各組の発生頻度を降順に並べることで前記番組特性の組を配列し、

前記新しさ順序付けは前記番組特性の組に含まれ、前記ユーザが以前に見た番組内で各組が発生した時を示す日付を降順に並べることで前記番組特性の組を配列し、

前記重み新しさ順序付けは前記頻度が同じである時は最も新しい日付を有する番組特性の組を優先すると同時に、前記発生頻度を降順に並べることで前記番組特性の組を配列することを特徴とする請求項36に記載の方法。

【請求項38】 前記関係の各々は、フィルタリングコマンド及び順序付けコマンドによって定義され、前記順序付けコマンドは重み係数順序付け、特殊性順序付け、一般性順序付け、新しさ順序付け、重み新しさ順序付け、特殊性重み順序付け、一般性重み順序付け、新しさ特殊性順序付け、特殊性新しさ順序付け及び一般性新しさ順序付けを含むリストから選択されることを特徴とする請求項26に記載の方法。

【請求項39】 前記重み係数順序付けは、前記ユーザが以前に見た番組内での各組の発生頻度を降順に並べることで前記番組特性の組を配列し、

前記特殊性順序付けは前記番組特性の組に含まれている番組特性の数に対応する特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記一般性順序付けは特殊性レベルを昇順に並べることで前記番組特性の組を配列し、

前記新しさ順序付けは前記番組特性の組に含まれ、前記ユーザが以前に見た番組内で各組が発生した時を示す日付を降順に並べることで前記番組特性の組を配列し、

前記重み新しさ順序付けは前記頻度が同じである時は最も新しい日付を有する番組特性の組を優先すると同時に、前記発生頻度を降順に並べることで前記番組特性の組を配列し、

前記特殊性重み順序付けは前記特殊性レベルが同じ時にはより高い発生頻度を有する番組特性の組を優先すると同時に、前記特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記新しさ特殊性順序付けは前記日付が同じ時にはより高い特殊性レベルを有する番組特性の組を優先すると同時に、前記日付を降順に並べることで前記番組特性の組を配列し、

前記特殊性新しさ順序付けは前記特殊性レベルが同じ時には最も新しい日付を有する番組特性の組を優先すると同時に、前記特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記一般性新しさ順序付けは前記特殊性レベルが同じ時には最も新しい日付を有する番組特性の組を優先すると同時に、前記特殊性レベルを昇順に並べることで前記番組特性の組を配列することを特徴とする請求項38に記載の方法。

【請求項40】 番組に関する情報が少なくとも1つの番組ガイド・リストを含むEPG（電子番組ガイド）データとして利用可能であるテレビ・システムで見る番組の推薦をユーザに行うための推薦システムであって、前記ユーザが以前に見た各番組に関連付けて複数の番組特性を記録する第1のメモリ手段と、

以前の番組要求に応じて行われた番組推薦に関して、複数の関係の各々と該関係に対応付けて前記ユーザの受諾の度合いとを記録する第2のメモリ手段と、

前記番組特性の組を形成するための処理手段と、前記関係から、前記受諾の度合いのうち最も高い度合いを有する少なくとも1つの関係を選択するタスク選択手段と、

前記選択された関係を使用して前記番組特性の組を順序付け、順序付けられた番組特性の組を生成する順序付け手段と、

前記選択された関係を使用して前記順序付けられた番組特性の組をフィルタリングし、フィルタリングされた番組特性の組を生成するフィルタリング手段と、

前記フィルタリングされた番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行い、最も高いフィルタリングされた番組特性の組を優先する検索手段と、

番組推薦を要求するユーザの入力で、前記フィルタリングされた組に最も適合する番組の有用性を番組推薦として前記ユーザに通知する画面上表示手段とを有することを特徴とする推薦システム。

【請求項41】 各組は、前記複数の番組特性を有することを特徴とする請求項40に記載の推薦システム。

【請求項42】 前記受諾の度合いを更新する更新手段を更に有することを特徴とする請求項40に記載の推薦システム。

【請求項43】 前記関係の各々が少なくとも第1のタ

タイプの関係と第2のタイプの関係の組合せを有し、前記第1のタイプの関係の前記度合いが延長された期間にわたって行われたユーザの選択の結果として更新され、前記第2のタイプの関係の前記度合いがより短い期間にわたって行われたユーザの選択の結果として更新されることを特徴とする請求項42に記載の推薦システム。

【請求項44】 前記関係の各々は、受諾の最大の度合いを有する前記第1のタイプの関係と、全ての第2のタイプの関係の組合せを有することを特徴とする請求項43に記載の推薦システム。

【請求項45】 前記関係は、各ユーザ独自に設定されることを特徴とする請求項40に記載の推薦システム。

【請求項46】 前記番組特性の組の各々は、前記ユーザが以前に見た番組内における各組の発生頻度に関連付けられていることを特徴とする請求項40に記載の推薦システム。

【請求項47】 前記番組特性の組の各々はそれに関連付けられた日付を有し、前記日付は前記ユーザが以前に見た番組内で各組が発生した時を示すことを特徴とする請求項40に記載の推薦システム。

【請求項48】 前記関係の各々は、フィルタリングコマンド及び順序付けコマンドによって定義されていることを特徴とする請求項40又は41に記載の推薦システム。

【請求項49】 前記フィルタリングコマンドは、時間スロット・フィルタ、重み係数フィルタ、人気フィルタ、緊急フィルタ、及びライフスタイル・フィルタを含むリストから選択されることを特徴とする請求項48に記載の推薦システム。

【請求項50】 前記時間スロット・フィルタは、推薦が要求される日又は時間にふさわしい日又は時間に依存し、前記日又は時間は前記番組特性の組に含まれ、各組が前記ユーザが以前に見た番組内で発生した時を示し、前記重み係数フィルタは前記ユーザが以前に見た番組内での各組の発生頻度に依存し、前記人気フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組が人気のあるイベント／番組に対応することを示す人気値に依存し、前記緊急フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組が緊急番組に対応することを示す緊急値に依存し、前記ライフスタイル・フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組がライフスタイルのタイプに対応することを示すライフスタイル値に依存することを特徴とする請求項49に記載の推薦システム。

【請求項51】 前記順序付けコマンドは、重み係数順序付け、新しさ順序付け、重み新しさ順序付けを含むリストから選択されることを特徴とする請求項48に記載の推薦システム。

【請求項52】 前記重み係数順序付けは、前記ユーザが以前に見た番組内での各組の発生頻度を降順に並べることによって前記番組特性の組を配列し、

前記新しさ順序付けは前記番組特性の組に含まれ、前記ユーザが以前に見た番組内で各組が発生した時を示す日付を降順に並べることで前記番組特性の組を配列し、前記重み新しさ順序付けは前記頻度が同じである時は最も新しい日付を有する番組特性の組を優先すると同時に、前記発生頻度を降順に並べることで前記番組特性の組を配列することを特徴とする請求項51に記載の推薦システム。

【請求項53】 前記関係の各々は、フィルタリングコマンド及び順序付けコマンドによって定義され、前記順序付けコマンドは重み係数順序付け、特殊性順序付け、一般性順序付け、新しさ順序付け、重み新しさ順序付け、特殊性重み順序付け、一般性重み順序付け、新しさ特殊性順序付け、特殊性新しさ順序付け及び一般性新しさ順序付けを含むリストから選択されることを特徴とする請求項41に記載の推薦システム。

【請求項54】 前記重み係数順序付けは、前記ユーザが以前に見た番組内での各組の発生頻度を降順に並べることで前記番組特性の組を配列し、

前記特殊性順序付けは前記番組特性の組に含まれている番組特性の数に対応する特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記一般性順序付けは特殊性レベルを昇順に並べることで前記番組特性の組を配列し、

前記新しさ順序付けは前記番組特性の組に含まれ、前記ユーザが以前に見た番組内で各組が発生した時を示す日付を降順に並べることで前記番組特性の組を配列し、前記重み新しさ順序付けは前記頻度が同じである時は最も新しい日付を有する番組特性の組を優先すると同時に、前記発生頻度を降順に並べることで前記番組特性の組を配列し、

前記特殊性重み順序付けは前記特殊性レベルが同じ時にはより高い発生頻度を有する番組特性の組を優先すると同時に、前記特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記新しさ特殊性順序付けは前記日付が同じ時にはより高い特殊性レベルを有する番組特性の組を優先すると同時に、前記日付を降順に並べることで前記番組特性の組を配列し、

前記特殊性新しさ順序付けは前記特殊性レベルが同じ時には最も新しい日付を有する番組特性の組を優先すると同時に、前記特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記一般性新しさ順序付けは前記特殊性レベルが同じ時には最も新しい日付を有する番組特性の組を優先すると同時に、前記特殊性レベルを昇順に並べることで前記番組特性の組を配列することを特徴とする請求項53に記載の推薦システム。

載の推薦システム。

【請求項55】 番組に関する情報を少なくとも1つの番組ガイド・リストを含むEPG（電子番組ガイド）として利用可能であり、

ユーザが以前に見た各番組に関連付けられた複数の番組特性と、以前の番組要求に応じて行われた番組推薦に関して、複数の関係の各々と該関係に対応付けられた前記ユーザの受諾の度合いとが記録されている、テレビ・システムで見る番組の推薦をユーザに行うためのコンピュータ・プログラムが記録されたコンピュータ読み取り可能な媒体を有するコンピュータ・プログラム製品であって、

前記番組特性の組を形成するためのコードと、

前記関係から、前記受諾の度合いのうち最も高い度合いを有する少なくとも1つの関係を選択するためのコードと、

前記選択された関係を使用して前記番組特性の組を順序付け、番組特性の順序付けられた組を生成するためのコードと、

前記選択された関係を使用して前記順序付けられた番組特性の組をフィルタリングし、番組特性のフィルタリングされた組を生成するためのコードと、

前記フィルタリングされた番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行い、最も高いフィルタリングされた番組特性の組を優先するためのコードと、

番組推薦を要求するユーザの入力で、前記フィルタリングされた組に最も適合する番組の有用性を番組推薦として前記ユーザに通知するためのコードとを有することを特徴とするコンピュータ・プログラム製品。

【請求項56】 各組は、前記複数の番組特性を有することを特徴とする請求項55に記載のコンピュータ・プログラム製品。

【請求項57】 前記受諾の度合いを更新するためのコードを更に有することを特徴とする請求項55に記載のコンピュータ・プログラム製品。

【請求項58】 前記関係の各々が少なくとも第1のタイプの関係と第2のタイプの関係の組合せを有し、前記第1のタイプの関係の前記度合いが延長された期間にわたって行われたユーザの選択の結果として更新され、前記第2のタイプの関係の前記度合いがより短い期間にわたって行われたユーザの選択の結果として更新されることを特徴とする請求項57に記載のコンピュータ・プログラム製品。

【請求項59】 前記関係の各々は、受諾の最大の度合いを有する前記第1のタイプの関係と、全ての第2のタイプの関係の組合せを有することを特徴とする請求項58に記載のコンピュータ・プログラム製品。

【請求項60】 前記関係は、各ユーザ独自に設定されることを特徴とする請求項55に記載のコンピュータ・

プログラム製品。

【請求項61】 前記番組特性の組の各々は、前記ユーザが以前に見た番組内における各組の発生頻度に関連付けられていることを特徴とする請求項55に記載のコンピュータ・プログラム製品。

【請求項62】 前記番組特性の組の各々はそれに関連付けられた日付を有し、前記日付は前記ユーザが以前に見た番組内で各組が発生した時を示すことを特徴とする請求項55に記載のコンピュータ・プログラム製品。

【請求項63】 前記関係の各々は、フィルタリングコマンド及び順序付けコマンドによって定義されていることを特徴とする請求項55又は56に記載のコンピュータ・プログラム製品。

【請求項64】 前記フィルタリングコマンドは、時間スロット・フィルタ、重み係数フィルタ、人気フィルタ、緊急フィルタ、及びライフスタイル・フィルタを含むリストから選択されることを特徴とする請求項63に記載のコンピュータ・プログラム製品。

【請求項65】 前記時間スロット・フィルタは、推薦が要求される日又は時間にふさわしい日又は時間に依存し、前記日又は時間は前記番組特性の組に含まれ、各組が前記ユーザが以前に見た番組内で発生した時を示し、前記重み係数フィルタは前記ユーザが以前に見た番組内での各組の発生頻度に依存し、

前記人気フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組が人気のあるイベント／番組に対応することを示す人気値に依存し、

前記緊急フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組が緊急番組に対応することを示す緊急値に依存し、

前記ライフスタイル・フィルタは前記番組特性の組に含まれ、前記ユーザが以前に見た各番組がライフスタイルのタイプに対応することを示すライフスタイル値に依存することを特徴とする請求項64に記載のコンピュータ・プログラム製品。

【請求項66】 前記順序付けコマンドは、重み係数順序付け、新しさ順序付け、重み新しさ順序付けを含むリストから選択されることを特徴とする請求項63に記載のコンピュータ・プログラム製品。

【請求項67】 前記重み係数順序付けは、前記ユーザが以前に見た番組内での各組の発生頻度を降順に並べることで前記番組特性の組を配列し、

前記新しさ順序付けは前記番組特性の組に含まれ、前記ユーザが以前に見た番組内で各組が発生した時を示す日付を降順に並べることで前記番組特性の組を配列し、

前記重み新しさ順序付けは前記頻度が同じである時は最も新しい日付を有する番組特性の組を優先すると同時に、前記発生頻度を降順に並べることで前記番組特性の組を配列することを特徴とする請求項66に記載のコンピュータ・プログラム製品。

【請求項68】 前記関係の各々は、フィルタリングコマンド及び順序付けコマンドによって定義され、前記順序付けコマンドは重み係数順序付け、特殊性順序付け、一般性順序付け、新しさ順序付け、重み新しさ順序付け、特殊性重み順序付け、一般性重み順序付け、新しさ特殊性順序付け、特殊性新しさ順序付け及び一般性新しさ順序付けを含むリストから選択されることを特徴とする請求項66に記載のコンピュータ・プログラム製品。

【請求項69】 前記重み係数順序付けは、前記ユーザが以前に見た番組内での各組の発生頻度を降順に並べることで前記番組特性の組を配列し、

前記特殊性順序付けは前記番組特性の組に含まれている番組特性の数に対応する特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記一般性順序付けは特殊性レベルを昇順に並べることで前記番組特性の組を配列し、

前記新しさ順序付けは前記番組特性の組に含まれ、前記ユーザが以前に見た番組内で各組が発生した時を示す日付を降順に並べることで前記番組特性の組を配列し、前記重み新しさ順序付けは前記頻度が同じである時は最も新しい日付を有する番組特性の組を優先すると同時に、前記発生頻度を降順に並べることで前記番組特性の組を配列し、

前記特殊性重み順序付けは前記特殊性レベルが同じ時にはより高い発生頻度を有する番組特性の組を優先すると同時に、前記特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記新しさ特殊性順序付けは前記日付が同じ時にはより高い特殊性レベルを有する番組特性の組を優先すると同時に、前記日付を降順に並べることで前記番組特性の組を配列し、

前記特殊性新しさ順序付けは前記特殊性レベルが同じ時には最も新しい日付を有する番組特性の組を優先すると同時に、前記特殊性レベルを降順に並べることで前記番組特性の組を配列し、

前記一般性新しさ順序付けは前記特殊性レベルが同じ時には最も新しい日付を有する番組特性の組を優先すると同時に、前記特殊性レベルを昇順に並べることで前記番組特性の組を配列することを特徴とする請求項68に記載のコンピュータ・プログラム製品。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明はテレビ番組の選択に関し、より詳細にはテレビ番組選択に推薦を提供するためのシステムに関する。

【0002】

【従来の技術】 テレビを見ることは多くの家庭では毎日行われ、スポーツ、映画、ニュース、事実の番組（例えば、ドキュメンタリやライフスタイル）などの幅広い番組内容に関して娯楽の源を提供している。

【0003】 従来、視聴者は地元の新聞又は雑誌に印刷されたテレビ・スケジュール一覧を見て、見るのに望ましい番組を見つけていた。更に、最近では、インターネット又はワールド・ワイド・ウェブを介して電子的な形で娯楽番組ガイドが利用可能になった。しかし、テレビ及びケーブル・テレビに衛星受信機が導入されたことにより、視聴者が利用可能なテレビ・チャンネルの数は劇的に増加した。これにより、印刷された番組一覧又は画面上の電子番組ガイド（EPG）を使用して見るテレビ番組を選択する作業が、非常に複雑で時間のかかるものになった。

【0004】 カテゴリ、時間又は役者を含む番組特性（特徴）によって順序付けられた番組一覧は、視聴者が選択を行う時に役に立つが、依然として時間のかかる作業である。典型的に、ばく大な数の利用可能な番組からその視聴者にとって興味のある番組はわずかしかないう可能性もある。

【0005】 視聴者が最も頻繁に見る番組のカテゴリを監視し、最も頻繁に見たカテゴリに基づいて推薦を提供するシステムが利用可能である。このような構成は、以前に見た番組から単一の番組特性（カテゴリなど）を選択し、この選択した番組特性に基づいて推薦を行うだけという短所をこうむっている。しかし、視聴者による選択は種々の番組特性及び番組特性の組合せにさえ依存する場合がある。更に、これらの番組特性及び番組特性の組合せを正確に結合して選択を行うために使用する方法は、各ユーザについて非常に個別的であり、従来技術のシステムはこのレベルの融通性を提供していない。

【0006】 本発明の目的は、既存の構成の1つ又は複数の欠点を実質的に克服すること、又は少なくとも改善することである。

【0007】

【課題を解決するための手段】 本発明の第1の態様によれば、番組のタイトル情報及び番組特性を少なくとも1つの番組ガイド・リストを含むEPG（電子番組ガイド）データとして利用可能なテレビ・システムで見る番組の選択を可能にする方法であって、前記テレビ・システムにおいてユーザが見た各番組と関連付けて複数の番組特性を記録するステップと、各組が前記番組特性の少なくとも2つからなる番組特性の組を形成するステップと、少なくとも各組を、特定の番組を見たいユーザの希望を表す順序付けられた値と関連付けるステップとを有し、番組推薦を要求するユーザの入力で、前記番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行い、前記番組特性の組に最も適合する番組の利用可能性を番組推薦として前記ユーザに通知することを特徴とする方法が提供される。

【0008】 本発明の別の態様によれば、番組のタイトル情報及び番組特性を少なくとも1つの番組ガイド・リストが含まれるEPG（電子番組ガイド）データとして

入力可能なテレビ・システムで見る番組の選択を可能にする推薦システムであって、前記テレビ・システムにおいてユーザが見た各番組と関連付けて複数の番組特性を記録するメモリ手段と、各組が前記番組特性の少なくとも2つを有する番組特性の組を形成し、少なくとも各組を、特定の番組を見たいユーザの希望を表す順序付けられた値と関連付ける処理手段と、前記番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行う検索手段と、番組推薦を要求するユーザの入力で、前記番組特性の組に最も適合する番組の有用性を番組推薦として前記ユーザに通知する画面上表示手段とを有することを特徴とする推薦システムが提供される。

【0009】本発明の更に別の態様によれば、番組のタイトル情報及び番組特性が少なくとも1つの番組ガイド・リストを含むEPG（電子番組ガイド）データとして利用可能なテレビ・システムで見る番組の選択を可能にするためのコンピュータ・プログラムが記録されたコンピュータ読み取り可能な媒体を有するコンピュータ・プログラム製品であって、前記テレビ・システムにおいてユーザが見た各番組と関連付けて複数の番組特性を記録するコンピュータ・プログラムコードと、各組が前記番組特性の少なくとも2つからなる番組特性の組を形成するコンピュータ・プログラムコードと、少なくとも各組を、特定の番組を見たいユーザの希望を表す順序付けられた値と関連付けるコンピュータ・プログラムコードとを有し、番組推薦を要求するユーザの入力で、前記番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行い、前記番組特性の組に最も適合する番組の有用性を番組推薦として前記ユーザに通知することを特徴とするコンピュータ・プログラム製品が提供される。

【0010】本発明の更に別の態様によれば、番組に関する情報を少なくとも1つの番組ガイド・リストを含むEPG（電子番組ガイド）として利用可能であり、ユーザが以前に見た各番組と関連付けられた複数の番組特性と、以前の番組要求に応じて行われた番組推薦に関して、複数の関係の各々と該関係に対応付けて前記ユーザの受諾の度合いとが記録されている、テレビ・システムで見る番組の推薦をユーザに行うための方法であって、前記番組特性の組を形成するステップと、前記関係から、前記受諾の度合いのうち最も高い度合いを有する少なくとも1つの関係を選択するステップと、前記選択された関係を使用して前記番組特性の組を順序付け、番組特性の順序付けられた組を生成するステップと、前記選択された関係を使用して前記順序付けられた番組特性の組をフィルタリングし、番組特性のフィルタリングされた組を生成するステップと、前記フィルタリングされた番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行い、最も高いフィ

ルタリングされた番組特性の組を優先するステップと、番組推薦を要求するユーザの入力で、前記フィルタリングされた組に最も適合する番組の有用性を番組推薦として前記ユーザに通知するステップとを有することを特徴とする方法が提供される。

【0011】本発明の更に別の態様によれば、番組に関する情報が少なくとも1つの番組ガイド・リストを含むEPG（電子番組ガイド）データとして利用可能であるテレビ・システムで見る番組の推薦をユーザに行うための推薦システムであって、前記ユーザが以前に見た各番組に関連付けて複数の番組特性を記録する第1のメモリ手段と、以前の番組要求に応じて行われた番組推薦に関して、複数の関係の各々と該関係に対応付けて前記ユーザの受諾の度合いとを記録する第2のメモリ手段と、前記番組特性の組を形成するための処理手段と、前記関係から、前記受諾の度合いのうち最も高い度合いを有する少なくとも1つの関係を選択するタスク選択手段と、前記選択された関係を使用して前記番組特性の組を順序付け、順序付けられた番組特性の組を生成する順序付け手段と、前記選択された関係を使用して前記順序付けられた番組特性の組をフィルタリングし、フィルタリングされた番組特性の組を生成するフィルタリング手段と、前記フィルタリングされた番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を行い、最も高いフィルタリングされた番組特性の組を優先する検索手段と、番組推薦を要求するユーザの入力で、前記フィルタリングされた組に最も適合する番組の有用性を番組推薦として前記ユーザに通知する画面上表示手段とを有することを特徴とする推薦システムが提供される。

【0012】本発明の更に別の態様によれば、番組に関する情報を少なくとも1つの番組ガイド・リストを含むEPG（電子番組ガイド）として利用可能であり、ユーザが以前に見た各番組に関連付けられた複数の番組特性と、以前の番組要求に応じて行われた番組推薦に関して、複数の関係の各々と該関係に対応付けられた前記ユーザの受諾の度合いとが記録されている、テレビ・システムで見る番組の推薦をユーザに行うためのコンピュータ・プログラムが記録されたコンピュータ読み取り可能な媒体を有するコンピュータ・プログラム製品であって、前記番組特性の組を形成するためのコードと、前記関係から、前記受諾の度合いのうち最も高い度合いを有する少なくとも1つの関係を選択するためのコードと、前記選択された関係を使用して前記番組特性の組を順序付け、番組特性の順序付けられた組を生成するためのコードと、前記選択された関係を使用して前記順序付けられた番組特性の組をフィルタリングし、番組特性のフィルタリングされた組を生成するためのコードと、前記フィルタリングされた番組特性の組に最も適合する番組特性を有する番組を見つけるためにEPGデータの検索を

行い、最も高いフィルタリングされた番組特性の組を優先するためのコードと、番組推薦を要求するユーザの入力で、前記フィルタリングされた組に最も適合する番組の有用性を番組推薦として前記ユーザに通知するためのコードとを有することを特徴とするコンピュータ・プログラム製品が提供される。

【0013】

【発明の実施の形態】図1Aに、多くの数の利用可能な番組から視聴者に適切な番組を自動的に提案するためのシステム50を示す。システム50は相互接続25を介して「セット・トップ」ボックス20に接続されたディジタル・テレビ(DTV)10を備えている。また、「セット・トップ」ボックス20は、ゲートウェイ41を介して外部のネットワーク42に接続されている。いくつかのコンテンツ・サーバ43及び電子番組ガイド(EPG)データベース22が、外部のネットワーク42に接続されている。コンテンツ・サーバ43は典型的にはコンテンツ・プロバイダによって提供され、映画及びテレビ番組を含むマルチメディア・コンテンツを含む。

【0014】利用可能な番組は、EPGデータベース22に一覧されている。コンテンツ・プロバイダの各々は関連付けられたEPGデータベース22を保持する可能性がある。EPGデータベース22内の一覧された利用可能な番組は、番組識別子により、コンテンツ・サーバ43内の対応するマルチメディア・コンテンツにリンクされる場合がある。典型的なEPGデータベース22からの抽出が図2に示されている。EPGデータベース22はいくつかの番組項目60を有し、番組項目60の各々は、値62を伴ういくつかの属性61を含む可能性がある。属性61には、次のようなものが含まれても良い。

- ・項目毎に唯一の番号であるEPG ID
- ・同じ番組IDを有する再実行に帰着する番組毎に唯一の番組ID
- ・例えば美術に対して「01」、ドラマに対して「16」、映画とシリーズ番組に対して「50」、スポーツに対して「75」などの、カテゴリID
- ・例えばアクション／冒険に対してサブカテゴリ「001」、コメディに対して「064」、犯罪に対して「074」などの映画カテゴリ「50」におけるサブカテゴリID
- ・タイトル
- ・番組項目に関する一般的なテキスト・フィールドである注
- ・キーワード
- ・評価
- ・EPGチャンネル
- ・開始曜日
- ・開始日

- ・開始時間
- ・長さ
- ・作成の年

DTVエージェント・システム21は、好ましくは「セット・トップ」ボックス20内で形成される。使用に際して、視聴者はリモコン30を使用してDTVエージェント・システム21と相互通信する。或いは、DTVエージェント・システム21はDTV10に統合されているか、図10に示されたようなパーソナル・コンピュータ100に組み込まれており、DTV10と適切にインタフェースされている。

【0015】好ましい実施形態では、DTVエージェント・システム21は、DTV10を制御するためのDTVエージェント36といくつかのアバタ・エージェント37を含み、各アバタ・エージェント37は特定の視聴者を表す。エージェント相互サーバ(IAS)35は、全てのエージェント相互の通信を管理し、またゲートウェイ41とインタフェースする。DTVエージェント・システム21は更に、アバタ・エージェント37によってアクセスされるいくつかの視聴者プロフィール・データベース23を含む。DTVエージェント36の機能には、次のようなものが含まれる。

- ・これによって視聴者に推薦が行われ、視聴者が選択を行うことを可能にする、DTV10の表示画面11を介した、グラフィカル・ユーザ・インタフェースの提供
- ・テレビ・チューナ・モジュール(図示せず)と相互作用して、複数の利用可能なテレビ・チャンネルから見るためのチャンネルを選択することを含む、DTV10の機能を制御すること
- ・視聴者の選択を集め、これらをアバタ・エージェント37に送達することアバタ・エージェント37の各々は、1人の視聴者及び視聴者プロフィール・データベース23のうち対応する1つに関連付けられている。図1Bに示すように、各アバタ・エージェント37は、特定のアバタ・エージェント37の制御を保持するアバタ・マネージャー38を含む。アバタ・マネージャー38はIAS35を介してEPGデータベース22にアクセスし、テレビ番組情報及び関連付けられた視聴者プロフィール・データベース23を得て、視聴者プロフィールを構築し保持する。また、アバタ・マネージャー38はIAS35を介してDTVエージェント36にメッセージを送信し、DTVエージェント36からメッセージを受信する役割も果たす。また、アバタ・エージェント37の中で、アバタ・マネージャー38は学習モジュール39及び推薦モジュール40の各々とインタフェースする役割も果たす。アバタ・エージェント37の1つのタスクは、視聴者によって行われ、学習モジュール39によって処理された選択に基づいて視聴者プロフィール・データベース23を構築し保持するものである。そして、アバタ・エージェント37は推薦モジュール40内で実

行される処理によって、視聴者プロフィール・データベース 2 3 を使用して視聴者に推薦を提供する。推薦は特定の視聴者に関して利用可能な、最も適切な視聴選択肢についてであり、表示画面 1 1 を介して視聴者に提供される。

【 0 0 1 6 】 図 1 A を再び参照すると、システム 5 0 は EPG データベース 2 2 及び I A S 3 5 の両方とインタフェースする EPG エージェント 4 4 を更に含んでも良い。EPG エージェント 4 4 は EPG データベース 2 2 にアクセスし、番組項目 6 0 から更なる属性を抽出し、属性には人気、緊急性、ライフスタイルが含まれる。人気属性は、番組項目 6 0 がフットボールのワールド・カップや人気のある演奏者によるコンサートなど、(現在) 人気のあるイベント／番組に対応するかどうかを示

す。緊急性属性は、番組項目 6 0 が特別なニュース／進捗報告、天気予報 (竜巻の警告など)、記者会見、国家演説などの緊急番組に対応するかどうかを示す。ライフスタイル属性は、番組項目 6 0 が昼間の映画、7 時のニュース、土曜日夜の音楽ショーなど一定のタイプのライフスタイルに対応することを示す。これらのさらなる属性の値は、真又は偽である。代替の実施形態では、これらの属性はすでに EPG データベース 2 2 に含まれている。

【 0 0 1 7 】 次に、表 1 及び表 2 を参照し、システム 5 0 の使用方法に関する典型的なステップの例を説明する。

【 0 0 1 8 】

【 表 1 】

表 1

ステップ	メッセージ	から	へ	内部処理
1#				視聴者は DTV10 のスイッチを入れる。
2				アニメーション・キャラクタ 12 が表示画面 11 上に表示される。視聴者はアニメーション・キャラクタ 12 を選択する。
3	SetAnimationSelection	DTVA	AA	
4#	SendUserData	AA	DTVA	
5				視聴者は表示画面 11 に表示されたプロンプトに基づいてインタラクティブに、個人情報と視聴履歴を提供する。
6	SetNewUserData	DTVA	AA	
*				
7#				視聴者は推薦 67 を要求する。
8				視聴者に時間と日付を求めるプロンプトが表示される。
9	SendRecommendationRequest	DTVA	AA	
*				
10#	SetRecommendations	AA	DTVA	
11				EPG およびレビュー・データが検索される。
12				レビュー・データは表示画面 11 上に表示される。視聴者は選択 68 を行う。
13	SetProgramSelection	DTVA	AA	
*				

エージェント37からのメッセージを待つ、「アイドル」モードに入る。このモードでは、DTV10はテレビ番組を表示する（視聴者によって最も最近選択されたチャンネル上に）。ハッシュ記号（#）によって示されたシーケンス・ポイントは任意の時にアクセスできる（即ち、非同期的に）。

【0020】表1はイベントのシーケンスと、DTVエージェント36から得られた結果のメッセージを示している。この中で、

1. ユーザはリモコン30を使用してDTV10を「オン」に入れる。

【0021】2. 視聴者は自分をDTVエージェント・システム21に識別する必要がある。

【0022】図3を参照すると、これはDTVエージェント36によって生成され表示画面11上に表示された一組のキャラクタからアニメーション・キャラクタ12を選択することによって行われても良い。その選択はリモコン30を使用して実行されても良い。好ましい実施形態では、キャラクタには3つのタイプがある。

(i) どの視聴者にも関連付けられていないキャラクタを意味する、触れられていないキャラクタ。

(ii) 視聴者に関連付けられているが、その特定の視聴者は現在関係していないキャラクタを意味する、眠っているキャラクタ。

(iii) 関連付けられた視聴者が現在選択されていることを識別する、アクティブなキャラクタ12。DTV10が「オン」にされたばかりでは、アクティブ・キャラクタは存在しない。

【0023】3. 識別された視聴者に関連付けられたアバタ・エージェント37のみを起動する、SetAnimationSelectionメッセージがアバタ・エージェント37に送信される。

【0024】4. DTVエージェント36に視聴者データを集めるように指示する、SendUserDataメッセージがアバタ・エージェント37から受信される。

【0025】5. DTVエージェント36は視聴者データの収集のためにDTV10の表示画面11を制御する。視聴者はリモコン30を使用してデータを提供する。キャラクタ12が初めて選択された時、システム50はこれがシステム50を使用する新しい視聴者であり、したがって登録されていない視聴者であると仮定する。アバタ・エージェント37は、現在登録されていない視聴者については、視聴者プロフィール・データベースを有しない。

【0026】そして、視聴者は再びリモコン30で選択を行うことによって現在経験されているいくつかのモードのうち1つのモードを選択するようにDTVエージェント36から要求されても良い。これらのモードの例としては幸せ、悲しい、どちらでもないなどを含む。番組のコンテンツがパスワードで保護されている時など、視

聴者からパスワードが要求される場合もある（例えば、アダルト・コンテンツ）。

【0027】6. 視聴者データはフォーマット化され、SetNewUserDataメッセージでアバタ・エージェント37に送信される。DTVエージェント36はアイドル・モードに入る。

【0028】7. 視聴者はリモコン30を使用してDTVエージェント・システム21から推薦されたテレビ番組の一覧を要求する。

【0029】8. DTVエージェント36は、視聴者に推薦が提供されるべき時間と日付のためのプロンプトを表示する。或いは、次の既定の分数内に開始するため、テレビ番組について推薦が自動的に行われても良い。

【0030】9. 時間と日付がフォーマット化され、SendRecommendationRequestメッセージでアバタ・エージェント37に送信される。DTVエージェント36はアイドル・モードに入る。

【0031】10. アバタ・エージェント37、及び特に推薦モジュール40は、視聴者の好みが所定の時間スロット及び視聴者のムードに最もマッチする、視聴するための視聴推薦をEPGデータベース22から決定する。アバタ・エージェント37はこれらの視聴推薦をSetRecommendationsメッセージでDTVエージェント36に送信する。

【0032】11. アバタ・エージェント37から受信された推薦された番組に基づいて、データがEPGデータベース22及び1つ又は複数のプレビュー・データベース（図示せず）から検索される。プレビュー・データベースはEPGデータベース22、又は別法としてはコンテンツ・サーバ43の一部を形成し、特定の番組の内容の簡単な例又はプレビューを提供しても良い。

【0033】12. DTVエージェント36は視聴推薦をフォーマット化し、表示画面11上にその推薦を表示する。図4に示すように、視聴推薦67は好ましくは、一組のプレビューを表示することにより、表示画面11上で視聴者に提供される。或いは、各々視聴推薦67を示すビットマップ画像又はタイトルのみが表示画面11上に表示される。また、各々の推薦67に関連付けられた追加情報も表示されて良い。また、視聴推薦67は番組属性を反映する方法で表示されても良い。例えば、共通のカテゴリを有する全ての視聴推薦67を一緒に表示する。視聴者は、リモコン30を使用して視聴推薦67を見て、視聴推薦67のうち1つを選択する。DTVエージェント36はDTV10と相互作用し、DTV10は選択された番組が放送される受信機チャンネルに回すことによって応答する。図5に示すように、選択された番組68は次いで、全表示画面11上で表示される。

【0034】13. 選択された番組68に関する情報は、SetProgramSelectionメッセージでアバタ・エージェント37に送信される。DTVエージェント36はア

アイドル・モードに入り、一方選択された番組 68 は表示画面 11 上に表示される。

【0035】表 2 は、イベントのシーケンスと、アバタ・エージェント 37 へ及びアバタ・エージェント 37 から結果として表示されるメッセージを示す。この中で 1. SetAnimationSelection メッセージは、選択されたアニメーション・キャラクタ 12 と関連付けられているアバタ・エージェント 37 によって DTV エージェント 36 から受信される。他のアバタ・エージェント 37 はアイドル・モードのままである。

【0036】2. アバタ・エージェント 37 は選択されたキャラクタ 12 が DTV エージェント・システム 21 に既知の視聴者に対応するか、又は新しい視聴者に対応するかどうかを判定する。視聴者が DTV エージェント・システム 21 に既知でない場合、次のシーケンス・ステップはステップ 3 である。視聴者が DTV エージェント・システムに既知の場合は、アバタ・マネージャー 38 はアイドル・モードに入る。

【0037】

【表 2】

表 2

ステップ	メッセージ	から	へ	内部処理
1#	SetAnimationSelection	DTVA	AA	
2				視聴者プロフィール・データベース 23 から情報を検索し、視聴者がプロフィール・データを有するかどうかを確立する。視聴者がプロフィール・データを有しない場合、ステップ 3 に行く。
*				
3				視聴者プロフィール・データベース 23 を作成する。
4	SendViewerData	AA	DTVA	
5	SetNewViewerData	DTVA	AA	
				視聴者プロフィール・データベース 23 を更新する。
7#	SendRecommendationRequest	DTVA	AA	
8				視聴者プロフィールおよび EPG 情報を、データベース 23 および 22 から検索する。
9				視聴者データおよびテレビ番組情報を推薦モジュール 40 に送信する。
10				推薦モジュール 40 は推薦されるテレビ番組 67 の一覧を構築する。
11				推薦モジュール 40 から番組推薦 67 を受信する。
12	SetRecommendations	AA	DTVA	
*				
13#	SetProgramSelection	DTVA	AA	
14				視聴者プロフィールおよび番組データを、データベース 23 および 22 から検索する。
15				視聴者データおよび新しい番組データを学習モジュール 39 へ送信する。
16				学習モジュール 39 はケース・ファイルを更新する。
17				ケース・ファイルを学習モジュール 39 から受信する。
18				視聴者プロフィール・データベース 23 を更新する。
*				

【0038】注：表 1 に示したシーケンスが*に達すると、アバタ・エージェント 37 は「アイドル」モードに入り、DTV エージェント 36 からのメッセージを待つ。このモードでは、学習モジュール 39 及び推薦モジュール 40 からのバックグラウンド処理が起きる場合が

ある。ハッシュ記号（#）によって示されたシーケンス・ポイントは任意の時（即ち、非同期的）にアクセスできる。

【0039】3. 新しい視聴者に関するプロフィールが視聴者プロフィール・データベース 23 内に作成され

る。

【0040】4. SendViewerDataメッセージがDTVエージェント36に送信される。

【0041】5. SetNewViewerDataメッセージがDTVエージェント36から受信される。このメッセージには静的な視聴者属性が含まれている。

【0042】6. アバタ・マネージャー38は、視聴者属性を視聴者プロフィール・データベース23に格納する。アバタ・マネージャー38はアイドル・モードに入る。

【0043】7. SendRecommendationRequestメッセージがDTVエージェント36から受信される。視聴者は特定の日及び時間に関する視聴者推薦67が提供されることを要求する。或いは、次の既定の分数内に開始するために、テレビ番組について推薦が自動的に行われても良い。

【0044】8. アバタ・マネージャー38は視聴者プロフィール・データベース23から視聴者プロフィール・データを検索し、テレビ番組情報をEPGデータベース22から検索する。データは推薦モジュール40で使用するようフォーマット化される。

【0045】9. 視聴者プロフィール・データ及び番組情報は、アバタ・マネージャー38によって推薦モジュール40に送信される。或いは、推薦モジュール40は直接データを検索する。

【0046】10. 推薦モジュール40は視聴者に推薦されるべき番組67のリストを構築する。

【0047】11. 推薦された番組67の一覧が推薦モジュール40からアバタ・マネージャー38へ送信される。

【0048】12. 推薦された番組67の一覧を含むSetRecommendationsメッセージがDTVエージェント36へ送信される。アバタ・マネージャー38はアイドル・モードに入る。

【0049】13. 視聴者が特定の番組68を選択したことを示す、SetProgramSelectionがDTVエージェント36から受信される。

【0050】14. 選択された番組68に関する番組項目60が、EPGデータベース22から検索される。また視聴者プロフィールは視聴者プロフィール・データベース23から検索される。

【0051】15. 選択された番組情報がケース・ファイルを更新するための学習モジュール39に送信される。

【0052】16. 学習モジュール39はケース・ファイルを更新し、選択された番組68を考慮する。

【0053】17. 更新されたケース・ファイルがアバタ・マネージャー38によって学習モジュール39から受信される。

【0054】18. 更新されたケース・ファイルは視聴

者プロフィール・データベース23を更新するために使用される。アバタ・マネージャー38はアイドル・モードに入る。

【0055】次に、学習モジュール39内の動作をより詳細に説明する。各番組はいくつの特徴(fi)と関連付けられており(ここで、特徴は本発明の請求の範囲に記載の番組特性に相当する)、各々の特徴は唯一の属性と属性値のペアを表す。その属性は、EPGデータベース22(図2に示されている)からの番組属性61と共に、選択が行われた時の視聴者属性を含んでも良い。視聴者属性は例えば、視聴者のムードを含む場合がある。各属性はいくつかの可能性のある値を有する。属性値ペアの例は、カテゴリ=「ドラマ」であり、カテゴリは属性で「ドラマ」は属性値である。

【0056】視聴者が行う各選択68はケース(Cj)と名付けられ、一組の特徴(fi)によって定義されている。例えば、ケースは(カテゴリ=「ドラマ」、サブカテゴリ=「コメディ・ドラマ」、…、EPGチャネル=「Nhk」、…)である。ケース(Cj)は利用可能な情報により、異なる数の特徴(fi)を有しても良い。全てのケースはケース・ファイル内の視聴者プロフィール・データベース23内に格納され、学習モジュール39に使用される。

【0057】学習モジュール39は、ケース・ファイルからのいくつかのケース(Cj)から特徴(fi)の共有されたパターン又は一般化パターンを識別するように動作する。学習モジュール39はケース・ファイルを入力として取り、一般化パターン・リスト(GPList)を生成し、これもまた視聴者プロフィール・データベース23内に格納される。そのGPListはケース・ファイルからの全ての一般化パターンを含む。GPList内の各一般化パターンは以下のように表される。

【0058】([交差]、発生) ここで、交差は異なるケース(Cj)によって共有される一組の特徴(fi)を示す。発生はこのような交差を共有するケースの数を示す。例えば、以下のケースから、

C1=(f1, f2, f3, f4, f7) ;

C2=(f1, f2, f5, f6, f7) ; 及び

C3=(f3, f5, f6, f8)

3つのケースから生成されるGPListは、3つの項目を有する。

【0059】({f1, f2, f7}, 2) ; ({f3}, 2) ; 及び ({f5, f6}, 2)

上記第1のGPList項目は、ケースC1及びC2の両方で発生する特徴f1, f2及びf7から生じる。他の項目は同様な方法で導出される。

【0060】新しいケース(Cj)が番組を見る視聴者による選択で利用可能になるたびに、これらの新しいケースはケース・ファイルに追加され、新しいGPListが学習モジュール39によって生成される。例えば、ケース・

ファイルは次の3つのケースを含むと仮定する。

【0061】C1=(カテゴリ=「ドラマ」、サブカテゴリ=「コメディ・ドラマ」、EPGチャネル=「Nhk」)、

C2=(カテゴリ=「ドラマ」、サブカテゴリ=「戦争」、EPGチャネル=「Nhk」、開始時間=「21:00」)、及び

C3=(カテゴリ=「スポーツ」、EPGチャネル=「Wow」、開始時間=「8:15」曜日=「土曜日」)

学習モジュールによって生成されるGPListは次の項目を含むのみである。

【0062】([カテゴリ=「ドラマ」、EPGチャネル=「Nhk」], 2) 従って、新しい視聴者選択68及び更なるケースが利用可能になると、例えば、

C4=(カテゴリ=「ドラマ」、サブカテゴリ=「トーク」、EPGチャネル=「Nhk」)、及び

C5=(カテゴリ=「スポーツ」、サブカテゴリ=「現在の事件」、開始時間=「10:00」、曜日=「土曜日」)

ケースC4及びC5はケース・ファイルに追加され、GPListは学習モジュール39によって計算し直され、次の項目を含むようになる。

【0063】([カテゴリ=「ドラマ」、EPGチャネル=「Nhk」], 3)

([カテゴリ=「スポーツ」、曜日=「土曜日」], 2) 学習モジュール39内の中心的な機能を実行するための手順MAINが、図7Aに示されている。3つのデータ構造、即ち、いくつかのケース(Cj)を含むケース・ファイル、全ての一般化パターンを保持するGPList、既に処理されたケースを含む検討されたケース・リストが手順において使用される。最初、視聴者が推薦を要求した時に手順MAINを開始すると、GPList及び検討されたケース・リストはからである。

【0064】ステップ200において手順MAINを開始し、ステップ201では新しいケースのリストをまず取得し、ステップ202では視聴者プロファイル・データベースからケース・ファイルを取得する。新しいケースを追加することによってケース・ファイルを更新するためにステップ203にてサブルーチンUPDATE-CASE-FILEを呼び出す。図7Bにおけるステップ212でサブルーチンUPDATE-CASE-FILEは開始される。

【0065】現在のGPListを持つケース・ファイルの各ケースのための交差を残るステップ204から211で取得する。ステップ204ではケース・ファイルから第1の項目を取り出し、ステップ205にてサブルーチンGEN-CASE-GPLISTを呼び出す時に入力ケースとして使用する。サブルーチンGEN-CASE-GPLISTは、入力ケースとGPList内の項目の間の一般化パターンを見つけ出し、図7Cにおけるステップ220で開始される。入力ケースと検討されたケース・リスト内のケースの間で一般化パターンを見つけ出すためにステップ206にてサブルー

チンGEN-CASE-EXAMINED-CASESを呼び出す。サブルーチンGEN-CASE-EXAMINED-CASESは、図7Hにおけるステップ310で開始される。ステップ207において、入力ケースを検討されたケース・リストに追加する。これによりケース・ファイルがからになるまでケース(Cj)はケース・ファイルから検討されたケース・リストへ連続的に移動される。ステップ208はケース・ファイルが更なる項目を有するかどうかを判定する。ケース・ファイルに残りの項目があるならば、手順MAINはステップ209に進み、ケース・ファイルから次の項目を入力ケースとして使用し、手順MAINはステップ205に続く。

【0066】ステップ208でケース・ファイルがからであることを判定した後、手順MAINはステップ210にて一般化パターンをGPList内に出力として生成し、ステップ211にて終了する。

【0067】ここで図7Bを参照してサブルーチンUPDATE-CASE-FILEを示す。このサブルーチンは手順MAINのステップ210において入力として生成された新しいケースを含み、現在のケース・ファイルにそれらを入れる。ケース・ファイルはステップ213において新しいケースのリストから項目を受信し、ステップ214において新しいケースのリストから次の新しいケースを取り出し、次の新しいケースをケース・ファイルの終わりに追加する。ステップ216で全ての新しいケースが処理されたかどうかを判定し、ステップ217及び215に進み、次の項目をケース・ファイルに追加する。ステップ219にて出力として更新済みケース・ファイルを生成し、サブルーチンUPDATE-CASE-FILEはステップ219にて終了する。

【0068】図7Cを参照すると、サブルーチンGEN-CASE-GPLISTは、入力ケースとGPList内の全ての項目の間の全ての一般化パターンを検出するものである。これは更に、入力ケースと現在のGPListの間の全ての新しい一般化パターンでGPListを更新する。

【0069】ステップ220においてGEN-CASE-GPLISTを開始し、ステップ221では入力として入力ケースと現在のGPListを受信する。ステップ222では現在のGPListが依然としてからであるかどうかを判定する。現在のGPListがからであるならば、入力ケースとGPListの間の一般化パターンは存在できず、サブルーチンはステップ236において戻る。

【0070】GPList内に項目があると、サブルーチンはステップ223に続き、サブルーチンG_List_GENを呼び出すことによって入力ケース及びGPList内の全ての項目の間の全ての一般化パターンを見つけ出す。入力ケースとGPListの間の可能性のある一般化パターンを含む、一般化パターンをG_Listに置く。サブルーチンG_List_GENは図7Dにおけるステップ240で開始される。

【0071】ステップ224では、G_Listがからであるかどうかを判定する。G_Listがからであるならば、入力

ケースとGPListの間の一般化パターンはサブルーチンG_List_GENによって見つけられず、サブルーチンGEN-CASE-GPLISTはステップ236において戻る。

【0072】G_List内に項目があると、サブルーチンGEN-CASE-GPLISTはステップ225に続き、サブルーチンUG_List_GENを呼び出す。サブルーチンUG_List_GENは図7Eにおけるステップ260で開始され、G_Listから唯一の一般化パターン・リスト、UG_Listを形成する。

【0073】ステップ226においてUG_Listから第1の項目、UG_Itemを検索し、UG_Itemから交差、First_Intersectionを抽出する。ステップ227においてGPListから第1の項目、GPList_Itemを検索し、GPList_Itemから交差、Second_Intersectionを抽出する。

【0074】ステップ228でサブルーチンINT_MATCHを呼び出し、First_IntersectionとSecond_Intersectionがマッチするかどうかを判定する。サブルーチンINT_MATCHは図7Fにおけるステップ280で開始される。ステップ228でFirst_IntersectionとSecond_Intersectionがマッチすることが分かったと、ステップ229にてUG_Itemの発生にGPList_Itemの発生を等しくする。サブルーチンはステップ233へ続く。

【0075】ステップ228でFirst_IntersectionとSecond_Intersectionがマッチしないことが分かったと、ステップ230でGPListの全ての項目が考慮されたかどうかを判定する。項目が残っていれば、ステップ231にてSecond_Intersectionとしてその交差と共にGPList内の次の項目を検索し、次にステップ228で再びFirst_IntersectionとSecond_Intersectionがマッチするかどうかを判定する。ステップ230でGPList内の全ての項目が考慮されたと判定したならば、ステップ232にてUG_ItemをGPListに追加し、サブルーチンはステップ233へ続く。

【0076】ステップ233でUG_List内の全ての項目が考慮されていないと判定すれば、ステップ234にてFirst_Intersectionとしてその交差と共にUG_List内の次の項目を検索し、ステップ227に続く。或いは、ステップ233でUG_List内の全ての項目が考慮されたと判定すれば、ステップ235にてサブルーチンGEN-CASE-GPLISTはGPListを出力し、ステップ236において戻る。

【0077】図7Dを参照して、入力ケースと、GPList内の全ての項目の間の全ての一般化パターンを決定するサブルーチンG_List_GENを説明する。ステップ240で開始し、入力として入力ケースとGPListを受信する。ステップ241において入力を得る。ステップ242においてGPListから第1の一般化パターン、GPList_Itemを取得し、ステップ243にて入力ケースから第1の特徴を検索し、ステップ244にてGPList_Itemの交差部分から第1の特徴を検索する。ステップ245で入力ケースから検索された特徴がGPList_Itemから検索された特

徴と同じであるかどうかを判定する。ステップ245でマッチが見いだされなければ、ステップ246でGPList_Itemから全ての特徴が処理されたかどうかを判定する。GPList_Item内に更なる特徴が残っているならば、ステップ255にてGPList_Itemから次の特徴を検索し、ステップ245へ続く。GPList_Itemの全ての特徴が考慮されたならば、サブルーチンはステップ247へ継続する。

【0078】ステップ245で肯定的な応答が得られると、ステップ252で新しい一般化パターンが作成されたかどうかを判定し、必要ならばステップ253にて新しい一般化パターンを1つ作成し、既に作成されていればステップ254へ進み、新しい一般化パターンに共有された特徴を追加する。サブルーチンはステップ247に続き、入力ケースから全ての特徴が処理されたかどうかを判定する。入力ケースの中に残っている特徴があるならば、ステップ256で入力ケースから次の特徴を検索する。

【0079】ステップ247で入力ケースから全ての特徴が考慮されたことを判定したならば、ステップ248で新しい一般化パターンがステップ252にて作成されたかどうかを判定する。一般化パターンがGPListの交差部分に既に存在していたならば、サブルーチンはステップ249に続き、新しい一般化パターンの発生に、同じ交差を有するGPList項目プラス1の値を与える。ステップ250において新しい一般化パターンをG_Listに追加する。

【0080】ステップ248で新しい一般化パターンがまだGPListに存在しないことを判定したならば、サブルーチンはステップ251へ継続する。

【0081】ステップ251で、GPListから全ての項目が考慮されたかどうかを判定する。GPList内に項目が残っていると、サブルーチンはステップ257に続き、ここでGPList内の次の項目を検索し、ステップ243が実行される。或いは、GPList内の全ての項目が考慮されたならば、サブルーチンG_List_GENはステップ258で新しいG_Listを出力として生成した後、ステップ259において戻る。

【0082】図7Eを参照して、唯一の一般化パターン・リスト、UG_Listを形成する手順であるサブルーチンUG_List_GENを示す。サブルーチンはステップ260で開始し、ステップ261では入力としてG_Listを受信する。ステップ262において、第1の一般化パターンをG_ListからUG_Listへコピーする。ステップ263ではG_Listから第1の項目、G_List_Itemを検索し、G_List_Itemから交差、First_Intersectionを検索する。ステップ264ではUG_Listから第1の項目である、UG_List_Itemを検索し、UG_List_Itemから交差、Second_Intersectionを検索する。

【0083】ステップ265はサブルーチンINT_MATCH

を呼び出し、First_IntersectionとSecond_Intersectionがマッチするかどうかを判定する。サブルーチンINT_MATCHは図7 Fにおけるステップ280で開始される。First_IntersectionとSecond_Intersectionがマッチすれば、2つの項目、G_List_Item及びUG_List_Itemのより高い発生を決定し、ステップ269及び270において、UG_List_Itemの発生として保存する。サブルーチンはステップ268に続く。

【0084】First_IntersectionとSecond_Intersectionがマッチしないと、ステップ266でUG_Listの全ての項目が考慮されたかどうかを判定する。項目が残っていれば、ステップ271にてSecond_Intersectionとしてその交差と共にUG_List内の次の項目を検索し、その後ステップ265で再びFirst_IntersectionとSecond_Intersectionがマッチするかどうかを判定する。ステップ266でUG_List内の全ての項目が考慮されたと判定したならば、続くステップ267にてUG_ListにG_List_Itemを追加し、サブルーチンはステップ268に続く。

【0085】ステップ268でG_List内の全ての項目が考慮されていないと判定すれば、ステップ263にてFirst_Intersectionとしてその交差と共にG_List内の次の項目を検索し、ステップ264に続く。或いは、ステップ268でG_List内の全ての項目が考慮されたと判定すれば、ステップ273にてサブルーチンUG_LIST_GENはUG_Listを出力し、ステップ274において戻る。

【0086】図7 Fを参照して、ステップ281で入力として得られたFirst_IntersectionとSecond_Intersectionが同じかどうかをチェックするサブルーチンINT_MATCHを示す。ステップ282でFirst_featureと名付けられたFirst_Intersectionからの第1の特徴を検索し、次にステップ283ではSecond_featureと名付けられたSecond_Intersectionから第1の特徴を検索する。ステップ284でサブルーチンFEATURE-SAMEを呼び出し、First_featureがSecond_featureと同じかどうかを判定する。サブルーチンFEATURE-SAMEは図7 Gにおけるステップ300で開始される。これらが同じであれば、次いでステップ285ではFirst_Intersectionの全ての特徴が考慮されたかどうかを判定する。考慮されていれば、サブルーチンはステップ286に続く。考慮されていなければ、ステップ296で次の特徴をFirst_Intersectionから検索し、First_featureと名付け、ステップ283に続く。しかし、ステップ284でFirst_featureがSecond_featureと同じでないと判定されたならば、ステップ290でSecond_Intersectionの全ての特徴が考慮されたかどうかを判定する。考慮されていれば、サブルーチンINT_MATCHはステップ292にて「NO」を戻す。Second_Intersectionの特徴が残っていれば、ステップ291で次の特徴をSecond_Intersectionから検索し、Second_featureと名付けてステップ284へ続く。

【0087】ステップ286で第1の特徴をSecond_Int

ersectionから検索し、Second_featureと名付け、次にステップ287に移り、ステップ287ではFirst_featureと名付けられたFirst_Intersectionから第1の特徴を検索する。ステップ288でサブルーチンFEATURE-SAMEを呼び出し、First_featureがSecond_featureと同じであるかどうかを判定する。それらが同じであれば、ステップ289でSecond_Intersectionの全ての特徴が考慮されたかどうかを判定する。考慮されていれば、サブルーチンINT_MATCHはステップ298にて「YES」を戻す。考慮されていなければ、ステップ297で次の特徴をSecond_Intersectionから検索し、第2の特徴と名付けてステップ287に続く。しかし、ステップ288でFirst_featureがSecond_featureと同じでないと判定されたならば、ステップ293でFirst_Intersectionの全ての特徴が考慮されたかどうかを判定する。考慮されていれば、サブルーチンINT_MATCHはステップ295にて「NO」を戻す。First_Intersectionの特徴が残っていれば、ステップ294で次の特徴をFirst_Intersectionから検索し、First_featureと名付けてステップ288へ続く。

【0088】さて図7 Gを参照して、2つの与えられた特徴が同じであるかどうかをチェックするサブルーチンFEATURE-SAMEを説明する。従って、ステップ301において、サブルーチンは入力としてFirst_featureとSecond_featureを受信する。ステップ302ではFirst-f-attributeと名付けられたFirst_featureの属性を取得し、ステップ303ではSecond-f-attributeと名付けられたSecond_featureの属性を取得する。ステップ304ではFirst-f-valueと名付けられたFirst_featureの値を取得し、ステップ305ではSecond-f-valueと名付けられたSecond_featureの値を取得する。

【0089】ステップ306でFirst-f-attributeがSecond-f-attributeと同じであるかどうかを判定し、ステップ307でFirst-f-valueがSecond-f-valueと同じであるかどうかを判定する。両方のステップ306及び307の答えが肯定的であれば、サブルーチンFEATURE-SAMEはステップ309にて「YES」を戻す。ステップ306及び307の答えの何れか1つが否定的であれば、サブルーチンFEATURE-SAMEはステップ308にて「NO」を戻す。

【0090】サブルーチンGEN-CASE-EXAMINED-CASESは図7 Hにおけるステップ310にて開始される。このサブルーチンは、ステップ311において入力として受信するケース・ファイルからの入力ケースと検討されたケース・リスト内のケースの間の一般化パターンを見つけ出す。ステップ312で、検討されたケース・リストからであるかどうかを判定し、肯定的であればステップ313において戻る。検討されたケース・リストが項目を持つならば、サブルーチンはステップ314に続き、検討されたケース・リストから第1のケースを検索す

る。ステップ315はサブルーチンGET-GEN-PATTERNを呼び出し、入力ケースと検討されたケース・リストからケースの間の一般化パターンGen-patternを計算する。サブルーチンGET-GEN-PATTERNは図7Iにおけるステップ330で開始される。

【0091】ステップ316でGen-patternが見つかったかどうかを判定する。Gen-patternが見つかったならば、サブルーチンはステップ317にてサブルーチンIF-MATCHを呼び出すことによってGPList内の任意の項目とマッチするかどうかを判定する。サブルーチンIF-MATCHは図7Jにおけるステップ350で開始される。Gen-patternがGPList内のどの項目にもマッチしなければ、ステップ318ではGen-patternを新しい項目としてGPListに追加し、ステップ319に続く。ステップ317でGen-patternがGPList内の項目にマッチすることを見つけ出すと、サブルーチンはステップ319に続き、検討されたケース・リストの全てのケースが考慮されたかどうかを判定する。

【0092】ステップ319で、検討されたケース・リスト内に考慮されるべきケースが残っていると判定すれば、ステップ320で、検討されたケース・リストから次のケースを検索し、ステップ315に続く。残っていないければ、ステップ321にてGPListを出力し、サブルーチンGEN-CASE-EXAMINED-CASESはステップ322において戻る。

【0093】図7Iを参照して、ケース・ファイルから入力ケースと検討されたケース・リストから1つのケースとの間の一般化パターン、Gen-patternを識別するためのサブルーチンGET-GEN-PATTERNを示す。サブルーチンはステップ330にて開始され、ステップ331にて入力ケースと検討されたケース・リストからケースを入力として取得する。サブルーチンはこれら2つのケースを入力として取り込み、その特徴を比較する。なんらかの特徴が2つのケースで共有されているならば、これらの共有された特徴はGen-patternの交差部分に含まれる。Gen-patternの発生は2となる。そうでなく、ケース間で共有された特徴がないならば、からのGen-patternが出力として生成される。

【0094】ステップ332でFirst_featureと名付けられた第1の特徴を入力ケースから検索し、次にステップ333に続き、ここでSecond_featureと名付けられた検討されたケース・リストからケースからの第1の特徴を検索する。ステップ334でサブルーチンFEATURE-SAMEを呼び出し、First_featureがSecond_featureと同じであるかどうかを判定する。サブルーチンFEATURE-SAMEは図7Gにおけるステップ300で開始される。それらが同じであれば、ステップ335はこの特徴をGen-patternの交差部分内に保存し、ステップ338に進む。ステップ334で特徴が同じでないと判定すれば、ステップ336で、検討されたケース・リストからケースの全

ての特徴が考慮されたかどうかを判定する。考慮されたならば、サブルーチンはステップ338に続く。考慮されていないければ、ステップ337で、検討されたケース・リストからケースの次の特徴を検索し、Second_featureと名付け、ステップ334に続く。ステップ338で入力ケースの全ての特徴が考慮されたかどうかを判定する。考慮されたならば、サブルーチンはステップ340に続く。しかし、ステップ338で入力ケースの全ての特徴が考慮されていないと判定したならば、ステップ339で入力ケースの次の特徴を検索し、First_featureと名付け、ステップ333に続く。

【0095】ステップ340でGen-patternがからであるかどうかを判定する。Gen-patternがからでなければ、Gen-patternの発生を2とし、サブルーチンGET-GEN-PATTERNはステップ342にて出力としてGen-patternを生成した後にステップ343において戻る。ステップ340でGen-patternがからであると判定すれば、サブルーチンGET-GEN-PATTERNはまたステップ343に戻る。

【0096】図7Jを参照して、サブルーチンIF-MATCHを示す。このサブルーチンは、Gen-patternの一般化パターンの交差がGPListの交差とマッチするかどうかをチェックするものである。サブルーチンはステップ350で開始され、ステップ351にてGen-patternとGPListを入力として受け入れる。

【0097】ステップ352でGen-patternの交差部分を抽出し、Gen-intersectionと名付ける。ステップ353ではGPListから第1の項目を検索し、ステップ354にてこの項目の交差部分を抽出してGP-intersectionと名付ける。

【0098】ステップ355でサブルーチンIF-SAMEを呼び出し、このサブルーチンは図7Kにおけるステップ370で開始され、Gen-intersectionとGP-intersectionが同じであるかどうかを判定する。マッチするものが見つけ出されると、サブルーチンIF-MATCHはステップ359にて「YES」を戻す。ステップ355でマッチするものが見つけ出されなかったならば、ステップ356でGPListの全ての項目が考慮されたかどうかを判定する。考慮されたならば、サブルーチンIF-MATCHはステップ360にて「NO」を戻す。しかし、ステップ356で考慮されていないGPListの項目があると判定されると、ステップ357でGPListから次の項目を検索し、ステップ358で項目の交差部分を抽出し、GP-intersectionと名付け、サブルーチンはステップ355に続く。

【0099】図7Kを参照して、ステップ371にて入力として得られたGen-intersectionとGP-intersectionが同じであるかどうかをチェックするサブルーチンIF-SAMEを示す。ステップ372でFirst_featureと名付けられた第1の特徴をGen-intersectionから検索し、次にステップ373ではSecond_featureと名付けられた、GP-i

ntersectionからの第1の特徴を検索する。ステップ374でサブルーチンFEATURE-SAMEを呼び出し、First_featureがSecond_featureと同じであるかどうかを判定する。サブルーチンFEATURE-SAMEは図7Gにおけるステップ300で開始される。それらが同じであるならば、ステップ378でGen-intersectionの全ての特徴が考慮されたかどうかを判定する。考慮されていれば、サブルーチンはステップ379へ続く。考慮されていなければ、ステップ388でGen-intersectionから次の特徴を検索し、First_featureと名付け、ステップ373へ続く。しかし、ステップ374でFirst_featureがSecond_featureと同じでないと判定したならば、ステップ375でGP-intersectionの全ての特徴が考慮されたかどうかを判定する。考慮されていれば、サブルーチンIF-SAMEはステップ292にて「NO」を戻す。GP-intersectionの特徴が残っていれば、ステップ376でGP-intersectionから次の特徴を検索し、Second_featureと名付け、ステップ374に続く。

【0100】ステップ379で第1の特徴をGP-intersectionから検索し、First_featureと名付け、次にステップ380ではGen-intersectionからSecond_featureと名付けられた第1の特徴を検索する。ステップ381でサブルーチンFEATURE-SAMEを呼び出し、First_featureがSecond_featureと同じかどうかを判定する。それらが同じであるならば、ステップ385でGP-intersectionの全ての特徴が考慮されたかどうかを判定する。考慮されていれば、サブルーチンIF-SAMEはステップ386にて「YES」を戻す。考慮されていなければ、ステップ387でGP-intersectionから次の特徴を検索し、First_featureと名付け、ステップ380に続く。しかし、ステップ381でFirst_featureがSecond_featureと同じでないと判定したならば、ステップ382でGen-intersectionの全ての特徴が考慮されたかどうかを判定する。考慮されていれば、サブルーチンIF-SAMEはステップ384にて「NO」を戻す。Gen-intersectionの特徴が残っ

ていれば、ステップ383で次の特徴をGen-intersectionから検索し、Second_featureと名付けてステップ381に続く。

【0101】図6に、特定の視聴者に関する視聴者プロフィール500の例を示す。視聴者プロフィール500は、図7A～7Kに示される方法によって決定され、特定の視聴者に関連付けられたアバタ・エージェント37の学習モジュール39によって実行されるGPListに対応し、次の交差と発生値を有するGPListを備える。

【0102】（[カテゴリ=「ドラマ」]，6）

（[カテゴリ=「ドラマ」，サブカテゴリ=「社会」，Year_of_make=「1999」]，3）

（[カテゴリ=「ドラマ」，サブカテゴリ=「社会」，Start_Time=「2200」，Year_of_make=「1999」]，2）

（[サブカテゴリ=「社会」]，8）

（[サブカテゴリ=「社会」，日=「火曜日」]，2）

（[カテゴリ=「映画」]，10）

（[サブカテゴリ=「社会」，日=「月」]，2）

（[カテゴリ=「映画」，サブカテゴリ=「社会」]，5）

（[サブカテゴリ=「社会」，日=「水曜日」]，2）

図6において、列501は交差の属性を表し、項目は属性の値502を表す。視聴者プロフィール500の各行503は視聴者プロフィール500内の項目を表す。また、視聴者プロフィール500の各項目503はDate Stamp項目を有する。好ましい実施形態では、項目503のDate Stamp値は、項目503の交差に貢献した最も新しい視聴者番組選択68の日付である。或いは、Date Stamp値は項目503内の交差に貢献した全ての視聴者番組選択68の日付の平均でも良い。例えば、第3行の項目503は、以下の値と解釈を有する。

【0103】

【表3】

属性	属性値	解釈
Category_ID	16	カテゴリ=「ドラマ」
Subcategory_ID	564	サブカテゴリ=「社会」
Start_Time	2200	Start_Time「2200」
Day	-1	考慮されない
EPG_channel	-1	考慮されない
Year_of_Make	1999	Year_of_make=「1999」

【0104】この交差の発生は2であり、この交差に貢献した番組選択68を視聴者が最後に行った時は1999年4月24日である。「-1」を伴う全ての項目は「考慮されない」と解釈される。

【0105】次に、推薦モジュール40内の動作をより詳細に説明する。好ましい実施形態によれば、推薦モジュール40はいくつかのタスクを使用して視聴者推薦6

7のリストをコンパイルする。種々のタスクが定義され、各タスクは視聴者プロフィール500の順序付けの方法の唯一の組み合わせと、順序付けられた視聴者プロフィールをフィルタリングするための特定の関連フィルタを定義する。

【0106】本実施形態によれば、各々のタスクの第1の様子は視聴者プロフィール500が順序付けられる方

法である。順序付け構成には次のようなものが含まれる。

【0107】・重み係数順序付けは、発生値を降順にすることによって視聴者プロフィール項目503を構成する、即ち、最も頻繁に発生する特徴 (fi) 交差が優先される。

【0108】・特殊性順序付けは、特殊性レベルを降順にすることによって視聴者プロフィール項目503を構成する、即ち、他より多い番組属性値ペアを特定する項目503が優先される。

【0109】・一般性順序付けは、特殊性レベルを昇順にすることによって視聴者プロフィール項目503を構成する、即ち、他より少ない番組属性値ペアを特定する項目503が優先される。

【0110】・新しさ順序付けは、Date_Stampを降順にすることによって視聴者プロフィール項目503を構成する、即ち、最も新しい項目503が優先される。

【0111】・重み新しさ順序付けは、発生値が同じであるときに、最も新しいDate_Stamp値を有する項目503を選ぶと同時に、発生値を降順にすることによって視聴者プロフィール項目503を構成する。

【0112】・特殊性重み順序付けは、特殊性レベルが同じであるとき、より高い発生値を有する項目503を選ぶと同時に、特殊性レベルを降順にすることによって視聴者プロフィール項目503を構成する。

【0113】・一般性重み順序付けは、特殊性レベルが同じであるとき、より高い発生値を有する項目503を選ぶと同時に、特殊性レベルを昇順にすることによって視聴者プロフィール項目503を構成する。

【0114】・新しさ特殊性順序付けは、Date_Stamp値が同じであるとき、より特殊な項目503を選ぶと同時に、Date_Stampを降順にすることによって視聴者プロフィール項目503を構成する。

【0115】以下の可能な順序付けも有益である。

【0116】・特殊性新しさ順序付けは、項目の特殊性レベルが同じであるとき、最も新しい項目503を選ぶと同時に、特殊性レベルを降順にすることによって視聴者プロフィール項目503を構成する。

【0117】・一般性新しさ順序付けは、項目の特殊性レベルが同じであるとき、最も新しい項目503を選ぶと同時に、特殊性レベルを昇順にすることによって視聴者プロフィール項目503を構成する。

【0118】本実施形態に応じて、各々のタスクの第2の様態は順序付けられた視聴者プロフィールに適用された特定の適切性フィルタであり、以下のものを含む。

【0119】・時間スロット・フィルタは、視聴者プロフィール項目503が時間特定又は日特定であるかどうかを確認し、特定の時間及び／又は日を推薦が要求された日及び時間であるかどうかを確認する。

【0120】視聴者プロフィール項目503を拒否する

条件：Start_Timeが視聴者プロフィール項目503内で特定されているが、推薦が要求された時間に等しくなかった場合、又は日が視聴者プロフィール項目503に特定されているが、推薦が要求された日に等しくなかった場合。時間スロット・フィルタは全てのフィルタ動作に含まれる。

【0121】・重み係数フィルタは、視聴者プロフィール項目503が全ての視聴者プロフィール項目503の現在の平均発生値より高い発生値を有するかどうかを確認する。

【0122】視聴者プロフィール項目503を拒否する条件：発生値 \leq 平均発生値。

【0123】・人気フィルタは、視聴者プロフィール項目503が（現在）人気のあるイベント／番組、例えば、フットボールのワールド・カップ、セリーヌ・ディオンのコンサートに対応するかどうかを確認する。

【0124】視聴者プロフィール項目503を拒否する条件：POPULARITY値が偽であるとき。

【0125】・緊急フィルタは、視聴者プロフィール項目503が緊急番組、例えば、特別なニュース／進捗状況報告、天気予報（竜巻の警告など）、記者会見、国家演説などに合っているかどうかを確認する。

【0126】視聴者プロフィール項目503を拒否する条件：URGENCY値が偽であるとき。

【0127】・ライフスタイル・フィルタは、視聴者プロフィール項目503が（既知の）タイプのライフスタイル、例えば、昼間の映画、7時のニュース、又は土曜日夜の音楽ショーに対応するかどうかを確認する。

【0128】視聴者プロフィール項目503を拒否する条件：LIFESTYLE値が偽であるとき。

【0129】フィルタ特性を組み合わせることによって、他のフィルタが得られても良い。例えば、

・人気ライフスタイル・フィルタは、視聴者プロフィール項目503が（既知の）タイプのライフスタイル及び（現在）人気のあるイベント／番組に対応するかどうかを確認する、例えば、土曜日夜のセリーヌ・ディオンのコンサートなどである。

【0130】このような組み合わせは非常に限定的であり、順序付けられた視聴者プロフィール項目を組み合わせられたフィルタでフィルタリングするタスクは、視聴者プロフィール項目503の全てではないが殆どを拒否する可能性がある。それゆえ、わずかな割合の視聴者（最も要求の多い視聴者）が、このようなフィルタを含むタスクを評価するであろう。

【0131】タスクのうちのいくつかは、より確立された視聴者プロフィール項目503（漸次的な変化）による推薦を提供するが、他のタスクは視聴者の視聴習慣の突然の変化に基づいて視聴者推薦を生成しようと試みる。

【0132】タスクの第1の様態（順序付け）及び第2

の態様（フィルタリング）の両方の組み合わせを使用し、以下のものを含むいくつかの有益なタスクが定義されても良い。

【0133】漸次変化タスク

基本タスク＝重み係数順序付け＋時間スロット・フィルタ（心の広い視聴者）

頑固な保守主義者タスク＝特殊性重み順序付け＋重み係数フィルタ（古い習慣の頑固な保守主義者）

昔のお気に入りタスク＝一般性重み順序付け＋重み係数フィルタ（オリジナルのレシピ）

最良選択タスク＝特殊性重み順序付け＋人気フィルタ（気難しい視聴者）

ゴールデンアワー・タスク＝重み新しさ順序付け＋ライフスタイル・フィルタ（私のテレビー私の城）

突然変化タスク

ホットトピック・タスク＝新しさ特殊性順序付け＋人気フィルタ（人気が変わったための突然変化）

新生活タスク＝新しさ特殊性順序付け＋ライフスタイル・フィルタ（ライフスタイルが変わったための突然変化）

監視タスク＝新しさ特殊性順序付け＋緊急フィルタ（危機的な状況による突然変化）

異なる視聴者が異なるタスクの実行の結果として生成された視聴者推薦に満足できるので、各タスクは関連付けられた視聴者特有のTask_Valueを有し、特定の視聴者が特定のタスクによって提供された視聴者推薦67に満足していることを示す。また、Task_Valueは視聴者プロフィール・データベース23に格納され、唯一のTask_IDによって表される各タスクは次のように格納される。

【0134】

【表4】

Task_ID	数字
Task_Value	数字
Task_Type	漸次的または突然の変化

【0135】一般に、視聴者推薦67の最適な組は、種々のタスク間の差の結果として得られる。好ましい実施形態では、漸次変化タスクと突然変化タスクとの組み合わせを使用して、視聴者推薦67をコンパイルする。

【0136】図8を参照すると、推薦モジュール40がアバタ・マネージャー38によって指示されテレビ番組推薦を提供すると、手順RECOMMENDはステップ801で開始される。手順RECOMMENDは入力として次のようなものを受信する。

【0137】・視聴者プロフィール・データベース23から視聴者プロフィール項目503を含む視聴者プロフィール500。各視聴者プロフィール項目503は交差する特徴と発生カウントを含む。

【0138】・視聴者推薦67を選択するための番組項目とそれらの特徴（fi）のリスト。このリストは視聴者推薦を要求する視聴者によって提供された時間／日付の組み合わせに基づいてアバタ・マネージャー38によって検索される。

【0139】・推薦された番組の所定の最大数。

【0140】最初、視聴者推薦のリストはからである。ステップ802で全ての視聴者プロフィール項目503が検討されるか、視聴者推薦の所定の最大数が既にあるか、番組項目のリストがからであるか、のいずれかが判定されると、手順はステップ805に続き、その視聴者推薦のリストはアバタ・マネージャー38に渡される。それ以外の場合は手順はステップ803に続き、ステップ803で次の視聴者プロフィール項目503を選択基準によって選択する。選択基準は以下の1つである。

【0141】・最も低い発生を伴う次の視聴者プロフィール項目503を選択する（即ち、より特定の似合いのものから開始する）。

【0142】・最も高い発生を伴う次の視聴者プロフィール項目503を選択する（即ち、最も一般的な似合いのものから開始する）。

【0143】・交差に最も多数の特徴を伴う次の視聴者プロフィール項目503を選択する（即ち、最も特定の似合いのもの）。

【0144】ステップ804では、ステップ803で選択された視聴者プロフィール項目503内に特徴（fi）がそれらを含む番組項目のリスト内の全ての番組項目60を番組項目のリストから視聴者推薦のリストへ移動する。手順はステップ802に戻る。

【0145】図9Aを参照して、テレビ番組推薦を提供するための代替の手順を示す。手順RECOMMENDはステップ601で開始される。ステップ602では手順RECOMMENDは入力として次のものを受信する。

【0146】・視聴者プロフィール項目503を含む、視聴者プロフィール・データベース23からの視聴者プロフィール500。各視聴者プロフィール項目503は交差する特徴、Date_Stamp及び発生カウントを含む；

・視聴者特定のTask_Values；

・EPGデータベース22から検索された番組項目とその特徴（fi）のリスト；及び

・推薦された番組の所定の所望の数R。

【0147】ステップ603で、カウンタ1は値1に設定される。変数Kも設定され、Kは推薦モジュールが視聴者推薦67を提供するために使用するタスクの数を表す。好ましい実施形態では、漸次変化タスクは最大のTask_Valueを有し、全ての突然変化タスクが使用される。従って、変数Kは突然変化タスクの数プラス1として設定される。

【0148】次いでステップ604が続き、そのTask_IDによって識別されたタスクを選択する。カウンタ1が

1 に等しいので、ステップ 6 0 4 で最大の Task_Value を伴う漸次変化タスクを選択する。カウンタ 1 が 1 より大きければ、次の突然変化タスクを選択する。

【0149】ステップ 6 0 5 で変数 $\alpha 1$ を設定する。変数 $\alpha 1$ は、推薦モジュール 4 0 がステップ 6 0 4 で選択されたタスクからなされたことを判定する視聴者推薦の所望の部分を表す。よって、変数 $\alpha 1$ は $1/(2n)$ として設定され、1 である n は漸次変化タスク、 $K-1$ である n は突然変化タスクをそれぞれ表す。

【0150】ステップ 6 0 6 で、選択されたタスクを適用し、視聴者推薦 6 7 を生成するためにサブルーチン REASON を呼び出す。サブルーチン REASON は、図 9 B のステップ 6 5 0 で開始される。

【0151】ステップ 6 0 7 で、REASON サブルーチンが選択されたタスクについて少なくとも 1 つの推薦を生成したかどうかを判定する。REASON サブルーチンが少なくとも 1 つの推薦を生成することができなかったならば、視聴者プロフィール項目 5 0 3 の交差する特徴をステップ 6 0 9 で緩める。好ましい実施形態では、カテゴリ及びサブカテゴリが交差として存在すれば、カテゴリ及びサブカテゴリを除いた全ての交差は無視される。手順 RECOMMEND はステップ 6 0 6 に戻り、ステップ 6 0 6 ではサブルーチン REASON が緩められた特徴と共に再び呼び出される。

【0152】ステップ 6 0 8 で視聴者プロフィール項目 5 0 3 の特徴が既に緩められていることを判定すれば、手順 RECOMMEND はステップ 6 2 1 に続く。

【0153】ステップ 6 0 7 で、REASON サブルーチンが選択されたタスクについて少なくとも 1 つの推薦を生成できたと判定すれば、ステップ 6 2 1 でカウンタ 1 が依然として変数 K より小さいかどうかを判定する。カウンタ 1 が値 K を有するならば、所望のタスクに関する推薦が生成され、手順 RECOMMEND はステップ 6 2 3 に続き、ステップ 6 2 3 で推薦された番組のリストを含む SetRecommendations メッセージがアバタ・マネージャ 3 8 に渡され、その SetRecommendations メッセージはアバタ・マネージャ 3 8 から DTV エージェント 3 6 に送信される。或いは、推薦モジュール 4 0 が SetRecommendations メッセージを直接 DTV エージェント 3 6 に送信しても良い。

【0154】しかし、ステップ 6 2 1 でカウンタ 1 が変数 K より小さいことを判定すれば、所定のタスクに関する視聴者推薦は依然として生成されなければならない。それゆえ、ステップ 6 2 2 でカウンタ 1 の値をインクリメントし、手順 RECOMMEND はステップ 6 0 4 に戻って次のタスクに関して視聴者推薦を判定する。

【0155】図 9 B を参照して、選択されたタスクを適用し、視聴者推薦を生成するためのサブルーチン REASON を示す。サブルーチン REASON はステップ 6 5 0 で開始される。

【0156】ステップ 6 5 2 では、手順 RECOMMEND のステップ 6 0 4 で選択されたタスクの順序付け態様を適用することによって視聴者プロフィール項目 5 0 3 を順序付ける。例えば、漸次変化タスク、頑固な保守主義者タスクが選択されたならば、視聴者プロフィール項目 5 0 3 は特殊性レベルが同じであるとき、より高い発生値を有する項目 5 0 3 を選ぶと同時に、特殊性レベルを降順にすることによって視聴者プロフィール項目 5 0 3 を構成する特殊性重み順序付けによって順序付けられる。ステップ 6 5 3 で、選択されたタスクによって特定されたフィルタを識別する。例えば、頑固な保守主義者タスクでは、ステップ 6 5 3 で視聴者プロフィール項目 5 0 3 が全ての視聴者プロフィール項目 5 0 3 の現在の平均発生値より高い発生値を有するかどうかを確認する、フィルタ重み係数フィルタを識別する。

【0157】ステップ 6 5 5 では視聴者プロフィール項目カウンタ j を値 1 に設定し、推薦カウンタ r を値 0 に設定する。ステップ 6 5 7 では j 番目に順序付けられた視聴者プロフィール項目 5 0 3 がステップ 6 5 3 から選択されたフィルタを満足させるかどうかを判定する。ここでも時間スロット・フィルタは常に全てのフィルタ動作に含まれ、これによって特定の Start_Time、Start_Day、又は Start_Date (即ち - 1 : 考慮しない、に等しくない) を伴い、特定の Start_Time、Start_Day 又は Start_Date が現在の時間、現在の曜日、現在の日付にそれぞれ等しくない、全ての視聴者プロフィール項目 5 0 3 が除去される。

【0158】ステップ 6 5 7 で j 番目の順序付けられた視聴者プロフィール項目 5 0 3 が選択されたフィルタを満足させないことを判定すれば、サブルーチン REASON はステップ 6 7 7 に続き、視聴者プロフィール・カウンタ j が依然として視聴者プロフィール項目 5 0 3 の合計数 N より小さいかどうかを判定する。

【0159】或いは、ステップ 6 5 7 で j 番目の順序付けられた視聴者プロフィール項目 5 0 3 が選択されたフィルタを満足することを判定すれば、サブルーチン REASON はステップ 6 6 0 に続き、ステップ 6 6 0 でサブルーチン BUILD_DYNAMIC_RULE を呼び出すことによって、ダイナミック・ルールが EPG データベース・フィルタとして構築される。サブルーチン BUILD_DYNAMIC_RULE は図 9 C に示され、ステップ 7 0 0 で開始される。

【0160】ステップ 6 6 2 ではサブルーチン BUILD_DYNAMIC_RULE 内で構築されたルールを EPG データベース 2 2 の項目 6 0 上に適用し、ルールを満足する EPG データベース項目 6 0 を識別する。これはサブルーチン INTERPRET を呼び出すステップ 6 6 5 に続く。サブルーチン INTERPRET は推薦のマップを構築し、図 9 D に示されている。

【0161】サブルーチン PRODUCE はステップ 6 6 7 で呼び出され、ステップ 6 6 7 で推薦のマップから視聴者

推薦の順序付けられたリストを生成する。サブルーチンPRODUCEは図9Eに示され、ステップ750で開始される。

【0162】ステップ672で推薦カウンタ r を、以前のカウンタとステップ667で呼び出されたサブルーチンPRODUCEで生成された推薦の数 m の合計として計算する。ステップ675で、推薦カウンタ r が推薦された番組の所定の所望の数 R の部分 $\alpha 1$ よりも小さいかどうかを判定する。ステップ675の判定が否定であれば、十分な数の推薦がそのタスクについて作成され、サブルーチンREASONはステップ682に続く。或いは、ステップ675で、作成された推薦が十分でないと判定すれば、サブルーチンREASONはステップ680に続き、ステップ680で順序付けられた推薦のリストをランク付けされた組 X に結合する。これは、推薦の順序付けられたリストから各項目に関して、項目がランク付けされた組 X に既に存在するかどうかを判定することによって行われる。項目が既に存在すれば、項目は無視され、次の項目が考慮される。しかし、項目がまだ存在しなければ、項目はランク付けされた組 X の最後に入力される。

【0163】次に、ステップ677で視聴者プロフィール・カウンタ j が依然として視聴者プロフィール項目503の合計より小さいかどうかを判定し、小さければ、サブルーチンREASONをステップ658に導き、カウンタ j をインクリメントしてからサブルーチンをステップ657に戻す。これは、フィルタ基準に合う次の視聴者プロフィール項目503を見つけ出すことにより、更なる推薦が決定されることを可能にする。

【0164】ステップ675から、或いはステップ677から、全ての視聴者プロフィール項目503が考慮されると、サブルーチンREASONはステップ682において手順RECOMMENDに戻る。

【0165】図9Cを参照して、サブルーチンBUILD_DYNAMIC_RULEを示す。サブルーチンBUILD_DYNAMIC_RULEは、選択されたフィルタを満足させた順序付けられた視聴者プロフィール項目503に関してダイナミック・ルールを構築し、ステップ700で開始される。

【0166】ステップ702でEPG番組データベース22のソースとして文字列 Y を設定する。文字列 Y 、従ってEPG番組データベース22はステップ705において文字列「tmp」で公式化される。ステップ707～712で変数の日、時間、分を、推薦が提供される日付と時間として設定する。この日付と時間はSendRecommendationRequestメッセージ内のDTVエージェントから得られる。変数の日、時間、分は文字列「tmp」に追加され、時間変数の周囲に2時間の時間帯を与える。

【0167】ステップ715で視聴者プロフィール項目503の交差する特徴が、ステップ609を実行することによって緩められたかどうかを判定する。サブルーチンBUILD_DYNAMIC_RULEは、視聴者プロフィール項目50

3の交差する特徴が緩められたならばステップ718に続き、ステップ718で、順序付けられた視聴者プロフィール項目503のカテゴリが-1（考慮しない）に等しくなければ、文字列「tmp」に追加される。同様に、ステップ719では、順序付けられた視聴者プロフィール項目503のサブカテゴリが-1（考慮しない）に等しくないと判定されたならば、文字列「tmp」に追加される。サブルーチンは次いでステップ720に進む。

【0168】ステップ715で交差する特徴が緩められていないことを判定すれば、ステップ717で順序付けられた視聴者プロフィール項目503の全ての属性が処理されたかどうかを判定する。そして、処理されたならば、ステップ720で文字列「rule」が文字列「tmp」と等しいように設定され、その後サブルーチンBUILD_DYNAMIC_RULEはステップ729でサブルーチンREASONに戻る。

【0169】しかし、順序付けられた視聴者プロフィール項目503に処理されていない属性があるならば、ステップ722及び725で値が-1（考慮しない）ではない次の属性を得てその属性を文字列「tmp」に追加する。サブルーチンはステップ717に続く。

【0170】次に、図9Dを参照して、サブルーチンINTERPRETを示す。サブルーチンINTERPRETは戻されたEPGデータベース項目60から推薦のマップを構築し、ステップ730で開始される。

【0171】サブルーチンINTERPRETはステップ732において視聴者推薦のマップをからに設定し、カウンタ c を1と設定することで初期化される。ステップ735では戻されたEPGデータベース項目60の c 番目の項目を取得し、ステップ737ではその項目を項目のカテゴリに従って推薦のマップに追加する。推薦のマップは項目としてカテゴリを有する行を有し、次に特徴として特定のカテゴリを有する全ての戻されたEPGデータベース項目60を有することになる。各行は典型的には異なる数のフィールドを有する。

【0172】ステップ740では、変数 c が戻されたEPGデータベース項目60の合計より小さいかどうか、言い換えれば全ての戻されたEPGデータベース項目60が推薦のマップに入力されたかどうかを判定する。ステップ740で全ての戻されたEPGデータベース項目60が追加されていないことを判定すれば、変数 c はステップ742でインクリメントされ、サブルーチンINTERPRETは再びステップ735へ続く。

【0173】しかし、ステップ740で全ての戻されたEPGデータベース項目60が推薦のマップに追加されたことを判定すれば、ステップ745ではサブルーチンINTERPRETをサブルーチンREASONに戻す。

【0174】図9Eを参照して、視聴者推薦を生成するためのサブルーチンPRODUCEを示す。サブルーチンPRODUCE

CEはステップ750で開始される。

【0175】サブルーチンPRODUCEはステップ752において、推薦のベクトルをからに設定し、カウンタhを1に設定することによって初期化される。ステップ755で推薦マップからh番目の行を取得し、ステップ757では推薦マップのh番目の行の第1の構成要素をカテゴリIDとして、及びそのカテゴリを伴う全ての戻されたEPGデータベース項目60（又はプログラム）としての第2の構成要素を特徴として検索する。ステップ758では、第1の構成要素及び第2の構成要素が加えられ、カテゴリ化された推薦のベクトルになる。

【0176】ステップ760で、変数hが推薦マップ内の行の合計数より小さいかどうか、言い換えれば、推薦マップ内の全ての行がベクトルに変換されたかどうかを判定する。ステップ760で推薦マップが完全には変換されていないことを判定すれば、変数hをステップ761でインクリメントし、サブルーチンPRODUCEは再びステップ755に続く。

【0177】しかし、ステップ760で推薦マップの全ての行が変換されていることを判定すれば、ステップ762でサブルーチンPRODUCEをサブルーチンREASONに戻す。

【0178】アバタ・マネージャー38が、視聴者が特定の番組68を選択したことを示すメッセージSetProgramSelectionをDTVエージェント36から受信すると（表2のステップ13）、アバタ・マネージャー38は図9Fに示されたルーチンON_PROGRAM_SELECTを呼び出し、ステップ770で開始される。ステップ775で、選択された番組が推薦Xのランク付けされたリストの前半内にあったかどうかを判定する。この判定が真であると分かれば、ステップ776でTask_Valueインクリメントである変数incを正の値に設定する。真でなければ、変数incをステップ777で負の値に設定する。

【0179】ステップ776又はステップ777で変数incを設定した後、ステップ782で変数incを手順RECOMMENDのステップ604で選択された漸次変化タスクのTask_Valueに追加する。従って、選択された番組がランク付けされた推薦Xの組の上半分にある場合、視聴者推薦67は成功したと見なされ、推薦モジュール40によって使用された漸次タスクのTask_Valueは増加され、これにより、視聴者推薦67は将来の推薦に再び使用されるようになる。しかし、選択された番組が一組のランク付けられた推薦Xの組の下半分にリストされた場合、推薦はおそらく最適ではなかったものであり、推薦モジュール40によって使用された漸次タスクのTask_Valueは減少される。

【0180】代替の実施形態では、サブルーチンINTERPRETは各推薦を生成するタスク（複数可）のTask_IDを記録する。これにより、ルーチンON_PROGRAM_SELECTが、選択された推薦を生成したタスクのTask_Valueを調節す

ることのみを可能にする。

【0181】図7A～7K、及び図8又は図9A～9Fの方法は、図10に示されたような従来の汎用コンピュータ・システム100を使用して実行され、このシステムでは、図7A～7K、図8及び図9のプロセスはコンピュータ・システム100内で実行するアプリケーション・プログラムなどのソフトウェアとして実装されても良い。特に、ステップは、コンピュータによって実行されるソフトウェア内の命令である。ソフトウェアは、後述する格納デバイスを含む、コンピュータ可読媒体内に格納されても良い。ソフトウェアはコンピュータ可読媒体からコンピュータにロードされ、次いでコンピュータによって実行される。このようなソフトウェア又はその上に記録されたコンピュータ・プログラムを有するコンピュータ可読媒体は、コンピュータ・プログラム製品である。コンピュータ内でコンピュータ・プログラム製品を使用することは、好ましくは本発明の実施形態によって有利な装置を実施する。

【0182】コンピュータ・システム100は、コンピュータ・モジュール102と、キーボード110及びマウス112などの入力デバイスと、プリンタ108及びディスプレイ・デバイス104を含む出力デバイスとを備える。

【0183】コンピュータ・モジュール102は、典型的には少なくとも1つのプロセッサ・ユニット114と、半導体ランダム・アクセス・メモリ（RAM）及び読取り専用メモリ（ROM）から形成されるメモリ・ユニット118と、ビデオ・インタフェース122を含む入力／出力（I/O）インタフェースと、キーボード110及びマウス112のためのI/Oインタフェース116とを含む。記憶装置124は典型的にはハード・ディスク・ドライブ126及びフロッピー（登録商標）・ディスク・ドライブ128を含み、提供される。磁気テープ・ドライブ（図示せず）が使用されても良い。CD-ROMドライブ120は典型的にはデータの非揮発性のソースとして提供される。コンピュータ・モジュール102の構成要素114～128は、典型的には相互接続バス130を介して通信し、当業者には公知のコンピュータ・システム100の動作の従来のモードにおける結果として生じる方法で通信する。この実施形態が実施できるコンピュータの例としては、IBM-PC及び互換のコンピュータ、それらから発展したSun Sparcstationsなどのコンピュータ・システムを含む。

【0184】典型的には、好ましい実施形態のアプリケーション・プログラムはハード・ディスク・ドライブ126に常駐され、その実行に際してプロセッサ114によって読み取られ、制御される。プログラムの中間記憶装置は半導体メモリ118を使用し、場合によっては、ハード・ディスク・ドライブ126と協力して達成されても良い。いくつかの例では、アプリケーション・プロ

グラムはCD-ROM又はフロッピー・ディスク上で符号化されて対応するドライブ120又は128を介して読み込まれて視聴者に提供されるか、或いはモデム・デバイス(図示せず)を介してネットワークから視聴者によって読み込まれても良い。更に、ソフトウェアは磁気テープ、ROM又は集積回路、光磁気ディスク、コンピュータ・モジュール102と別のデバイス間の無線送信チャネル又は赤外線送信チャネル、PCMCIAカードなどのコンピュータ可読カード、及び電子メール送信及びウェブサイトなどに記録された情報を含むインターネット及びイントラネットを含む、別のコンピュータ可読媒体からコンピュータ・システム100にロードすることもできる。上述したものは適切なコンピュータ可読媒体の例に過ぎない。他のコンピュータ可読媒体も本発明の範囲と精神から逸脱することなく実行可能である。

【0185】或いは、図7Aから図7K、図8及び図9Aから図9Fの方法は、図7Aから図7K、図8及び図9Aから図9Fの機能又はサブ機能を実行する1つ又は複数の集積回路などの専用ハードウェアにおいて実施されても良い。このような専用ハードウェアはグラフィック・プロセッサ、デジタル信号プロセッサ、1つ又は複数のマイクロプロセッサ及び関連付けられたメモリを含んで良い。

【0186】以上の説明は本発明のいくつかの実施形態のみを説明したものであり、本発明の範囲と精神を逸脱することなく修正及び／又は変更をそこに行うことができ、実施形態は例示的なものであって、限定的なものではない。

【図面の簡単な説明】

【図1A】システムの概念図である。

【図1B】図1Aにおけるシステムのアバタ・エージェントの詳細な表示である。

【図2】典型的な電子番組ガイドからの抽出である。

【図3】図1Aにおけるシステムのディスプレイ画面に表示された、アニメーション化されたキャラクタの図である。

【図4】図1Aにおけるシステムによって推薦がどのように視聴者に提供されるかの例である。

【図5】図4における推薦から行われた選択の例である。

【図6】視聴者プロフィール・データベースの例である。

【図7A】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7B】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7C(a)】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によ

って行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7C(b)】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7D(a)】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7D(b)】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7E】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7F(a)】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7F(b)】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7G】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7H】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7I】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7J】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7K(a)】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図7K(b)】図1Aに例示されたアバタ・エージェントの学習モジュールによって実行される、視聴者によって行われた選択から共通の特徴を学ぶ方法の流れ図である。

【図8】図1Bに例示されたアバタ・エージェントの推薦モジュールによって実行される、学習モジュールからの結果を最良に論理付け、視聴者に見る番組の推薦を行う方法の流れ図である。

【図9A】学習モジュールからの結果を最良に論理付

け、視聴者に見る番組の推薦を行う代替の方法の流れ図である。

【図9B】学習モジュールからの結果を最良に論理付け、視聴者に見る番組の推薦を行う代替の方法の流れ図である。

【図9C】学習モジュールからの結果を最良に論理付け、視聴者に見る番組の推薦を行う代替の方法の流れ図である。

【図9D】学習モジュールからの結果を最良に論理付け、視聴者に見る番組の推薦を行う代替の方法の流れ図

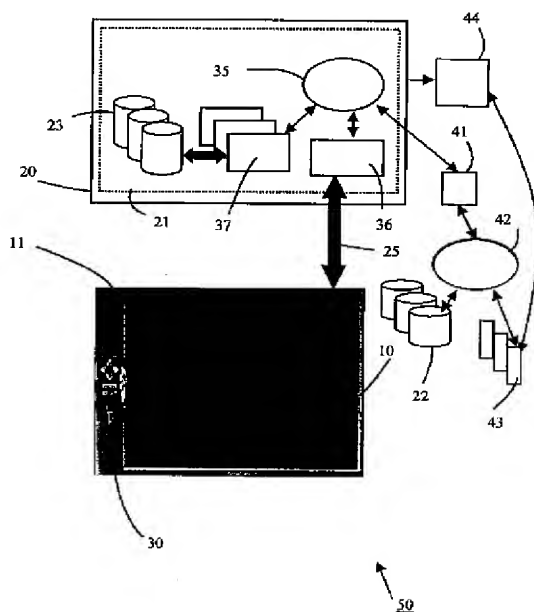
である。

【図9E】学習モジュールからの結果を最良に論理付け、視聴者に見る番組の推薦を行う代替の方法の流れ図である。

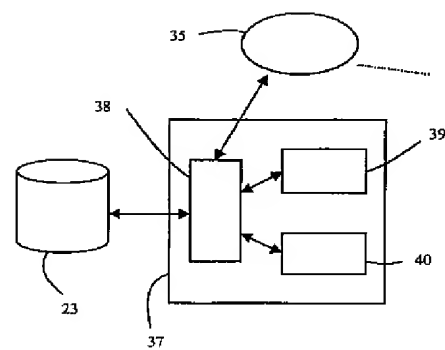
【図9F】学習モジュールからの結果を最良に論理付け、視聴者に見る番組の推薦を行う代替の方法の流れ図である。

【図10】説明された構成が実行できる汎用コンピュータの概念構成図である。

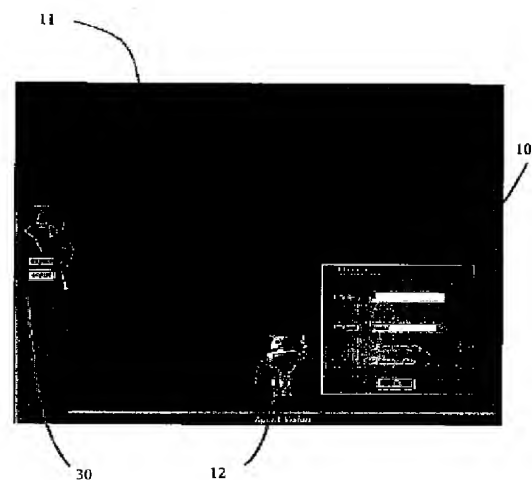
【図1A】



【図1B】



【図3】



【 図 2 】

EPG ID	プログラム ID	サブカテゴリ ID	タイトル	EPG チャネル	曜日	日付	開始時間	正確な長さ	作成の年
1	1	16	564 ferishte (felicity) no seishun	Wow	Fri	4/9/99	20:00	50	1999
2	2	16	564 Suzuran	Nhk	Mon	4/5/99	8:15	15	1999
3	3	16	217 genroku ryoran	Nhk	Sun	4/4/99	20:00	45	1999
4	4	16	564 nichiyō gekijō/goddō nyū-su	Tbs	Sun	4/11/99	21:00	54	1999
5	5	16	563 abunai ho-kago	Asahi	Mon	4/12/99	20:00	54	1999
6	6	16	564 rippu suteikku (lip stick)	Fuji	Mon	4/12/99	21:00	54	1999
7	7	16	519 Romansu	Nihon	Mon	4/12/99	22:00	84	1999
8	8	16	188 furehata ninzaburo	Fuji	Tue	4/13/99	21:00	69	1999
9	9	16	001 kizudarake no onna	Fuji	Tue	4/13/99	22:15	54	1999
10	10	16	564 happi ai to kando no monogatari	Tokyo	Wed	4/14/99	20:00	54	1999
11	11	16	563 Naomi	Fuji	Wed	4/14/99	21:00	54	1999
12	12	16	564 aka ga hashiru - to-gei seishun ki	Nhk	Wed	4/17/99	22:00	45	1999
13	13	16	519 semi daburu	Fuji	Wed	4/14/99	22:00	54	1999
14	14	16	188 Maiko san wa meitantei!	Asahi	Wed	4/15/99	20:00	54	1999
15	15	16	519 koi no kiseki	Asahi	Thu	4/15/99	21:00	69	1999

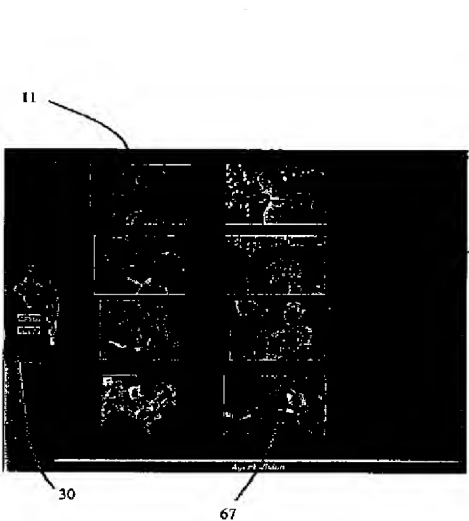
61

22

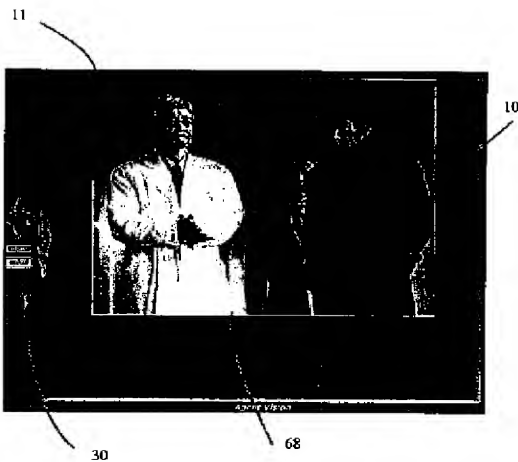
60

62

【図4】



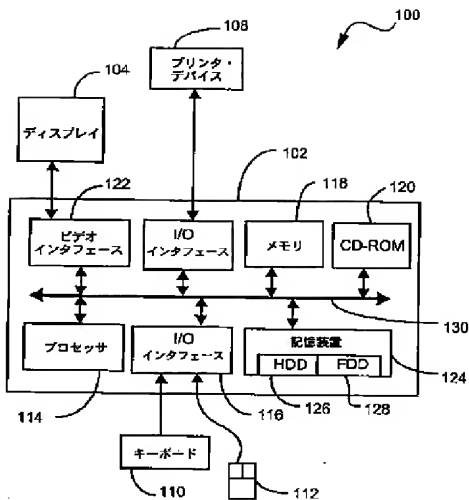
【図5】



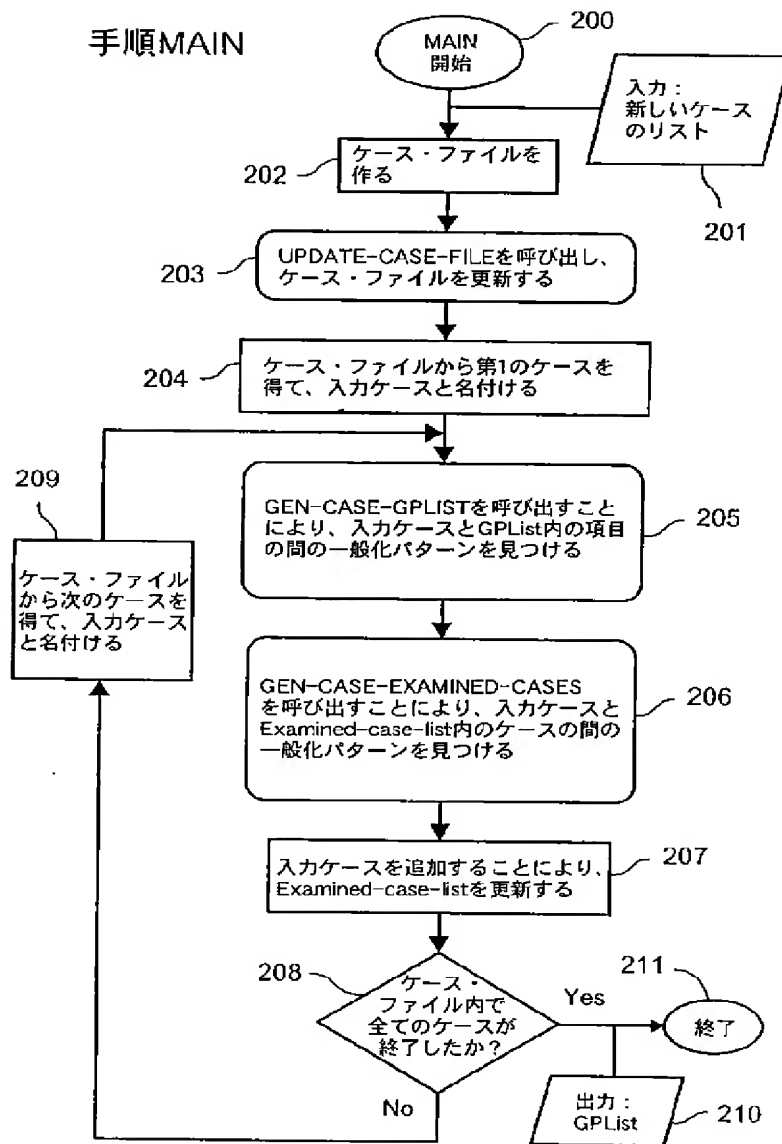
【図6】

Category_ID	Subcategory_ID	Start_Time	曜日	EPG_channel	Year_of_make	発生
16	-1	-1	-1	-1	-1	6
16	564	-1	-1	-1	1999	3
16	564	2200	-1	-1	1999	2
-1	564	-1	-1	-1	-1	8
-1	564	-1	5	-1	-1	2
50	-1	-1	-1	-1	-1	10
-1	564	-1	2	-1	-1	2
50	564	-1	-1	-1	-1	5
-1	564	-1	4	-1	-1	2

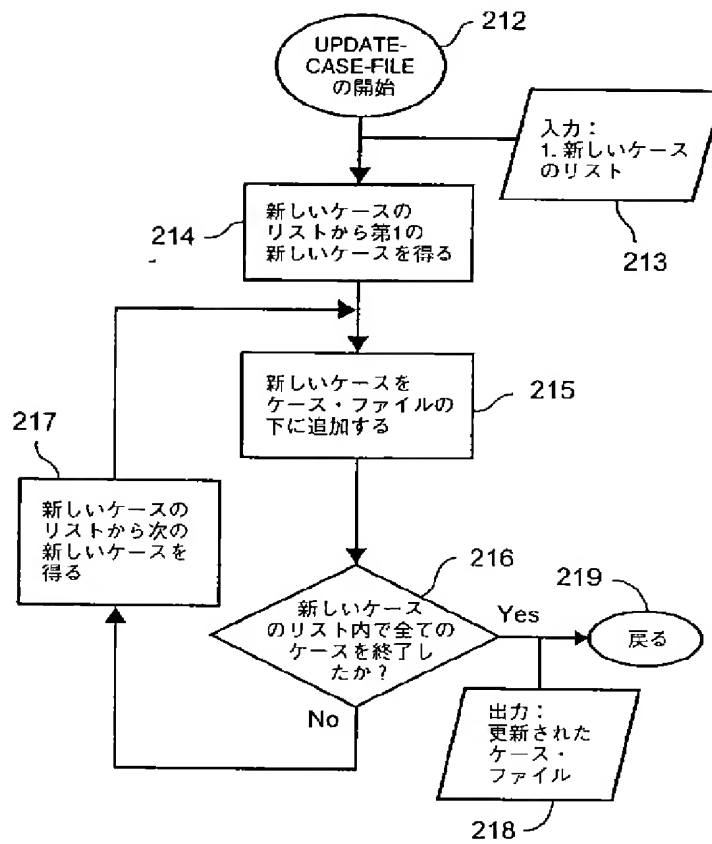
【図10】



【図7A】

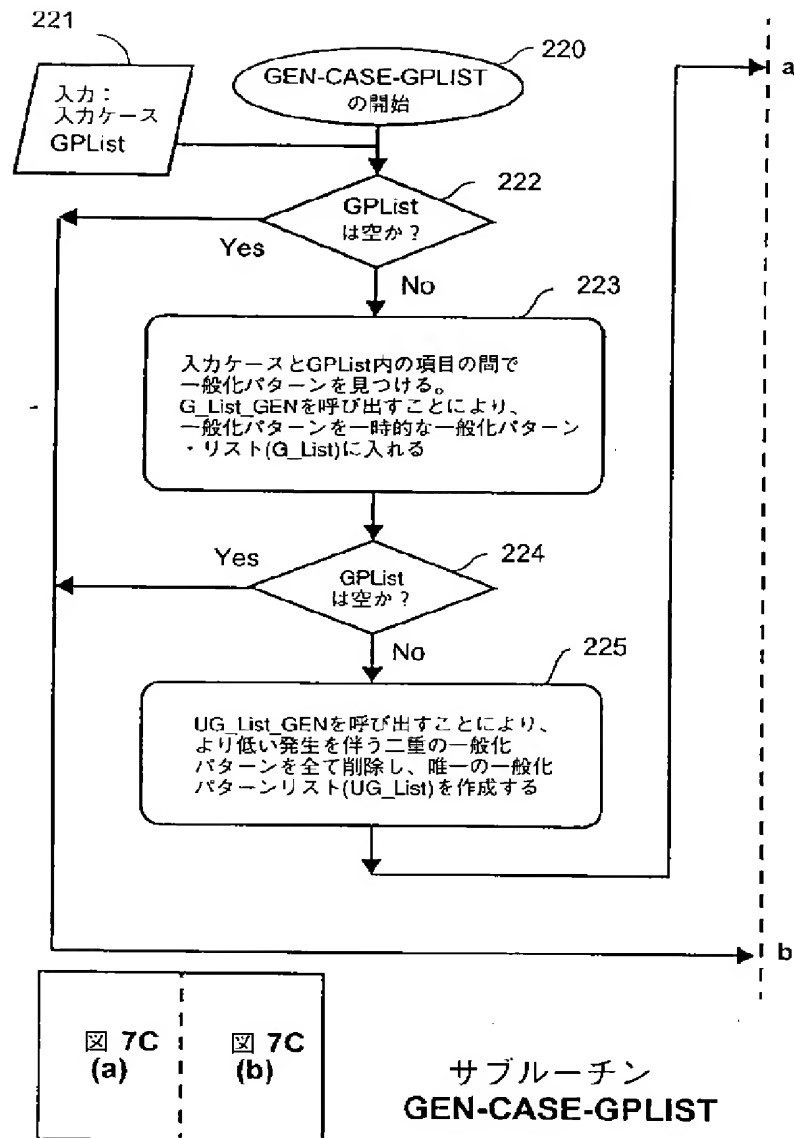


【図7B】



サブルーチンUPDATE-CASE-FILE

【図7C(a)】



【 図 7 C (b) 】

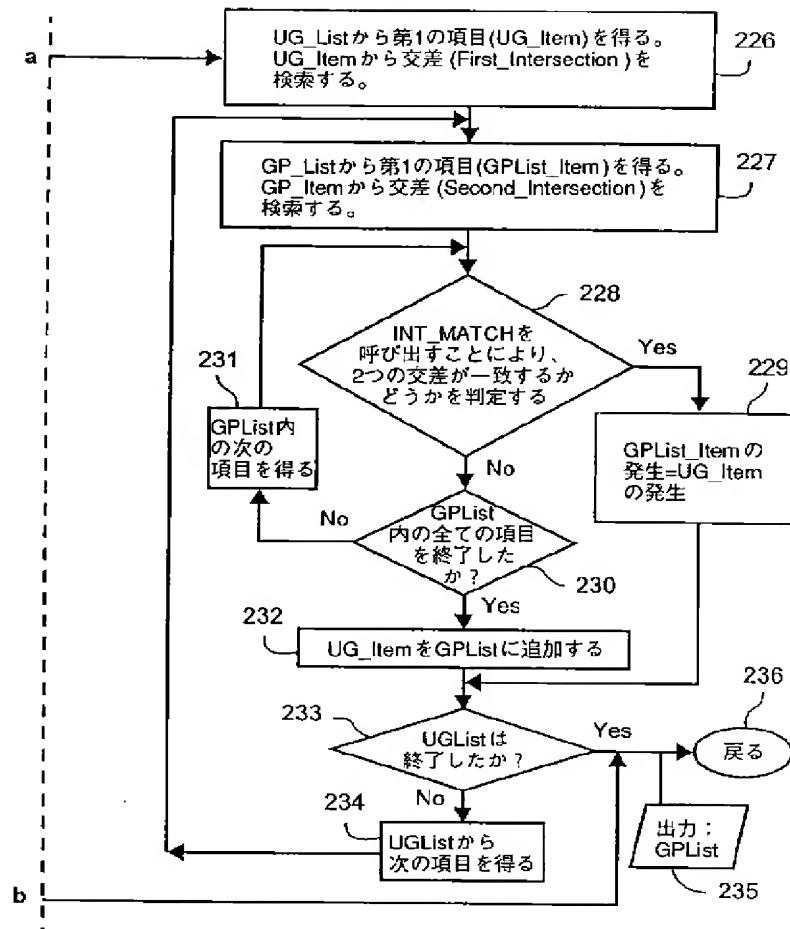
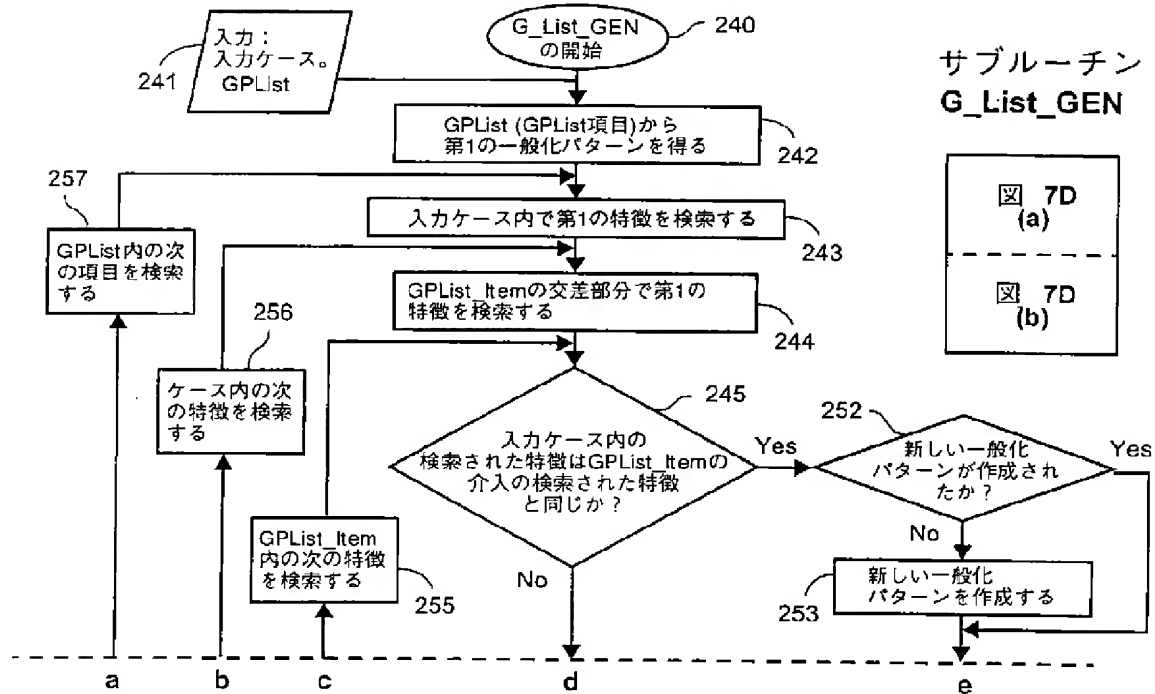


図 7C
(a)

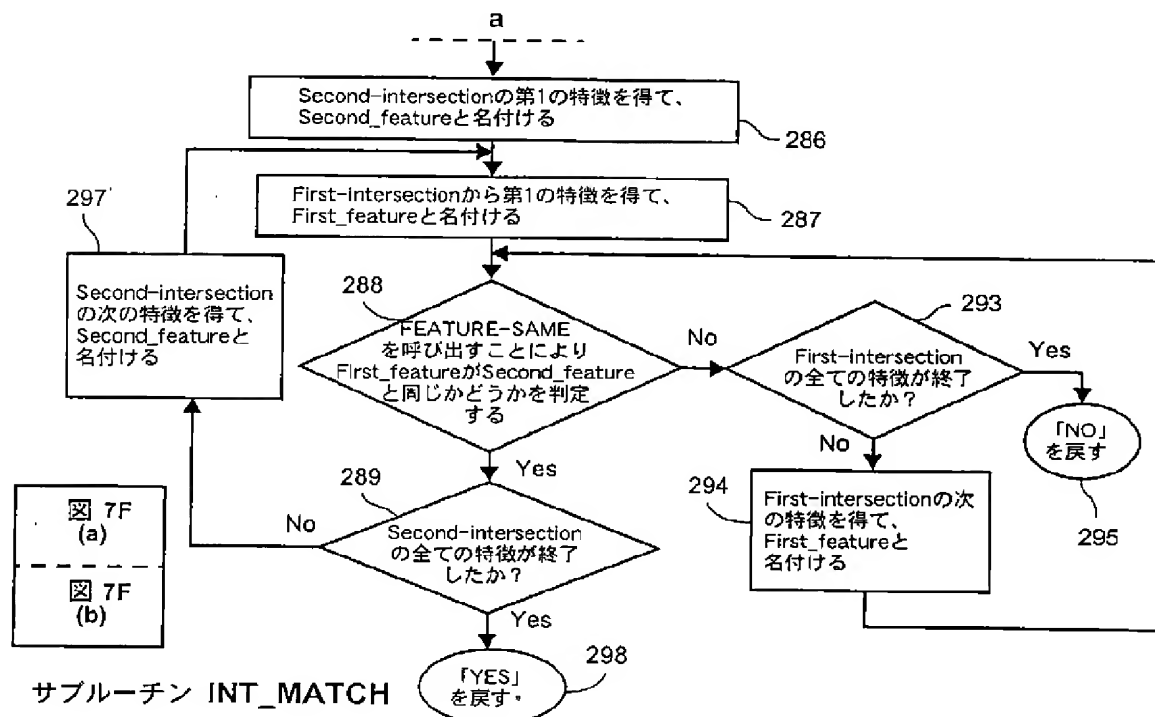
図 7C
(b)

サブルーチン
GEN-CASE-GPLIST

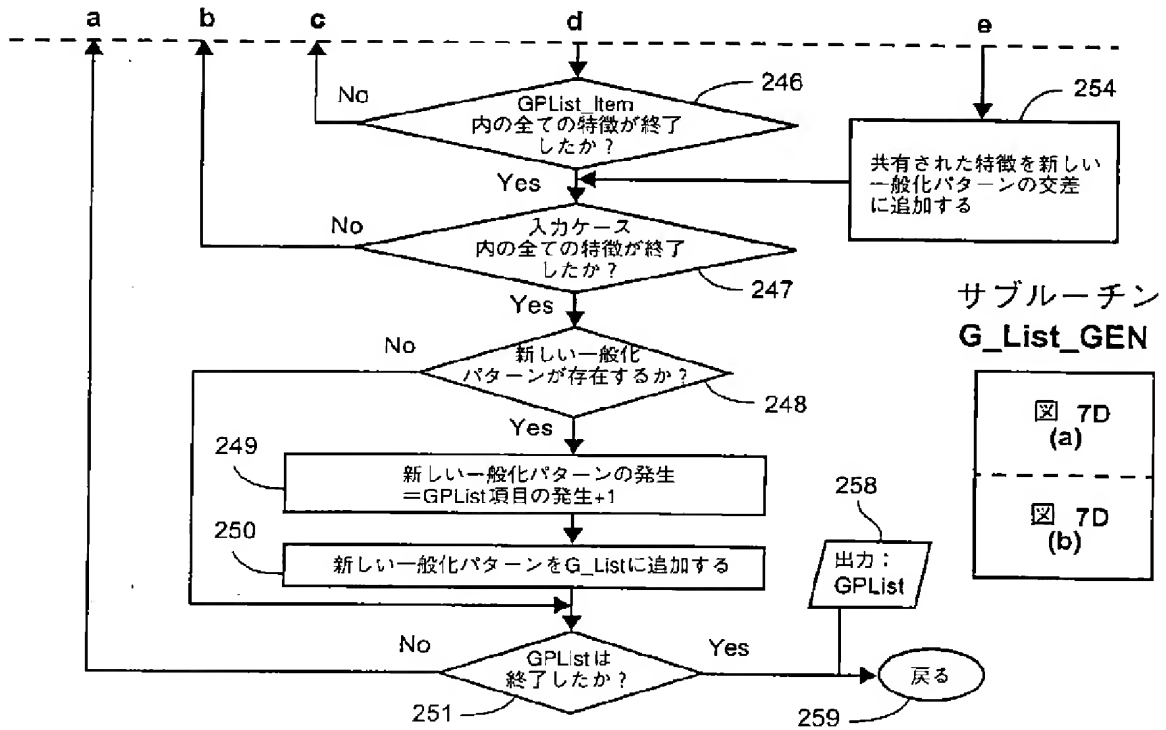
【図7D (a)】



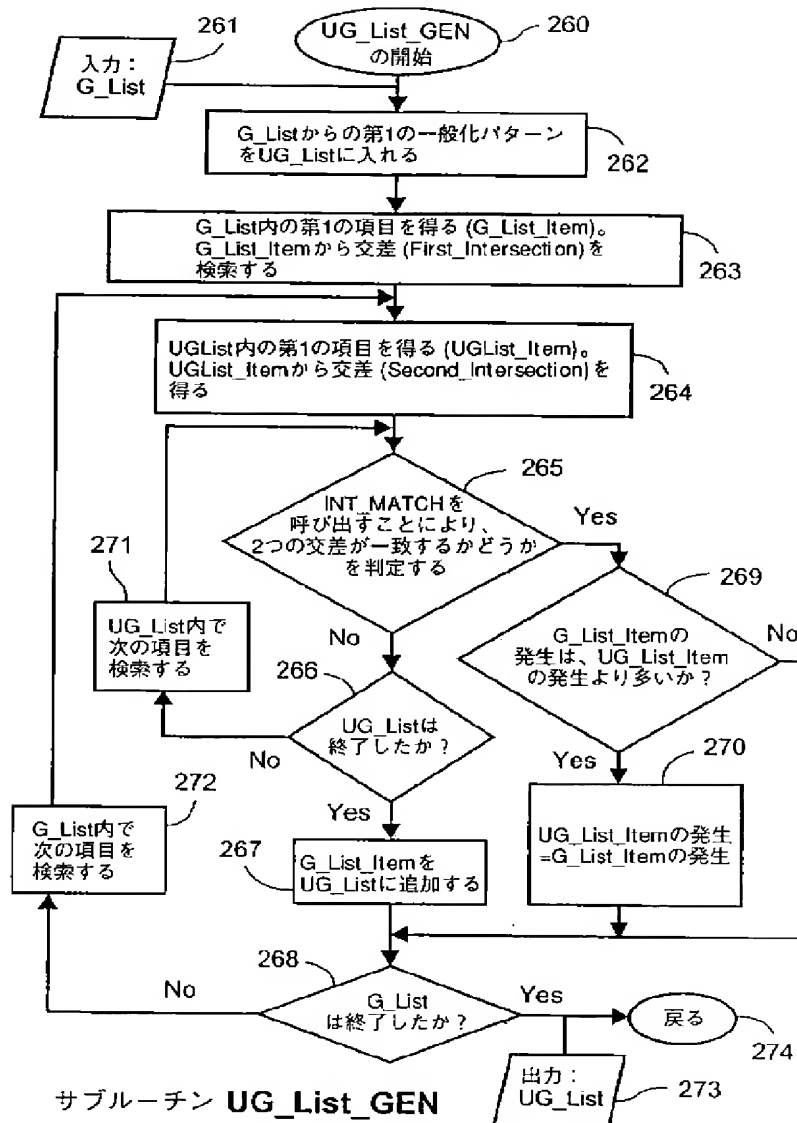
【図7F (b)】



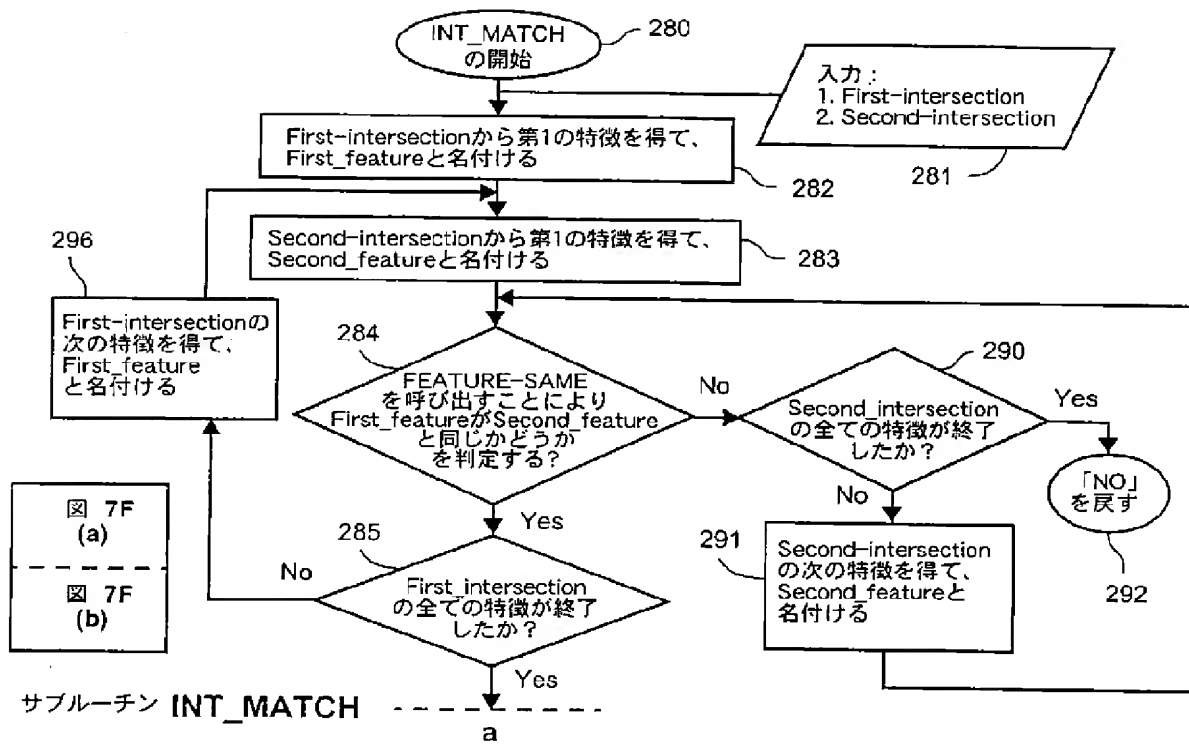
【図7D (b)】



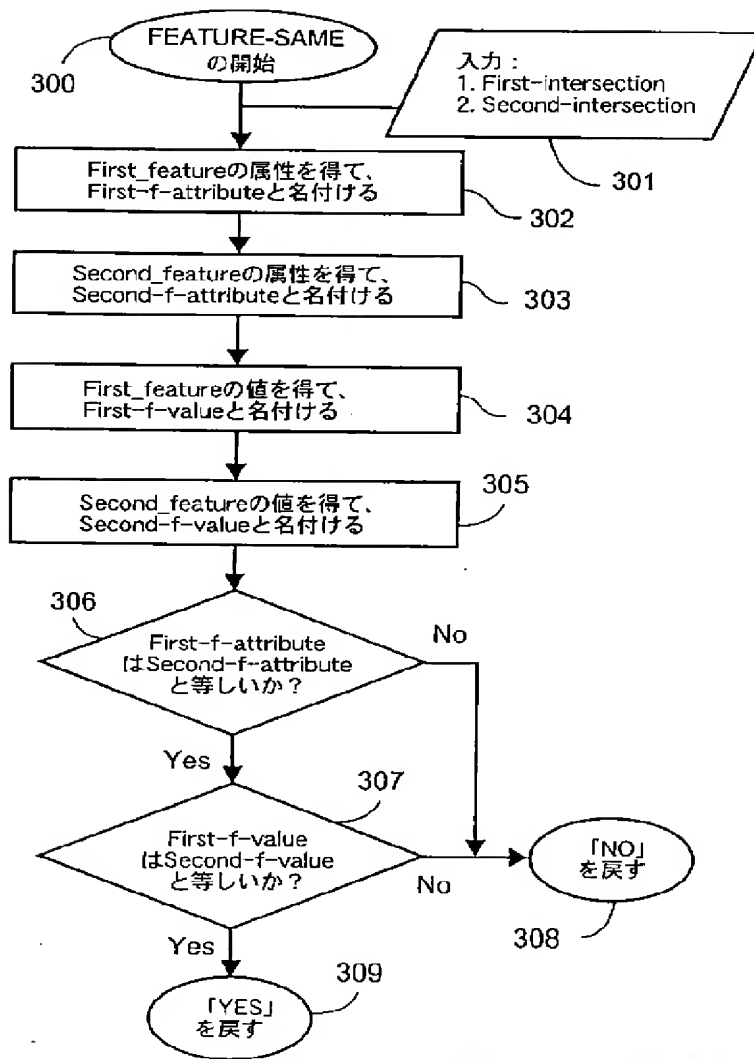
【図7E】



【図7F(a)】

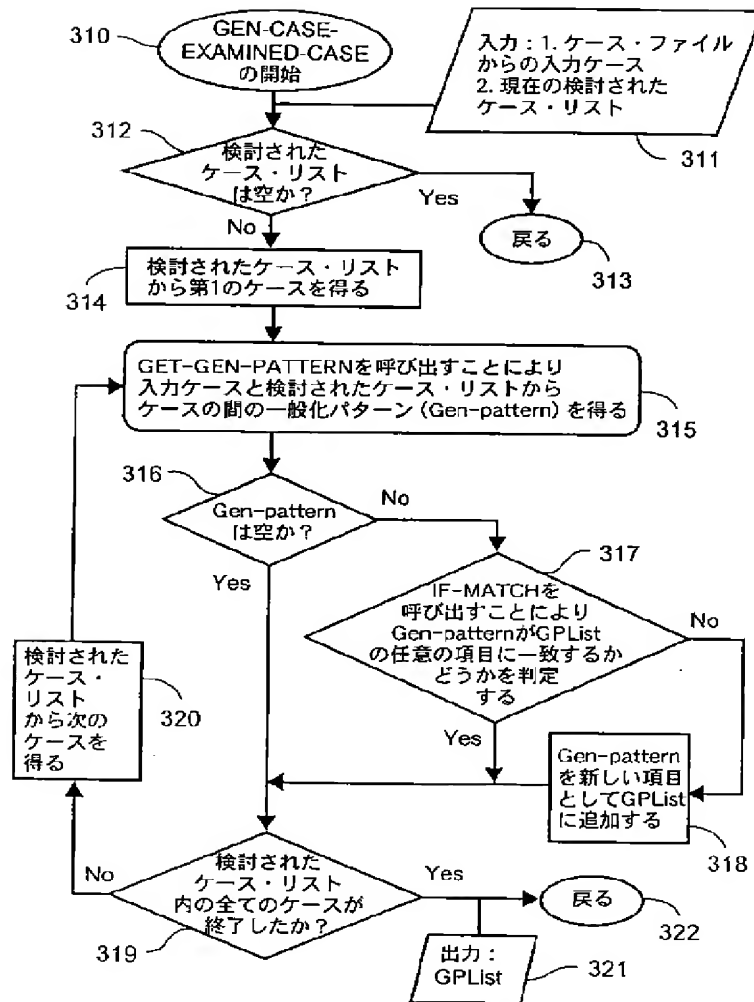


【図7G】



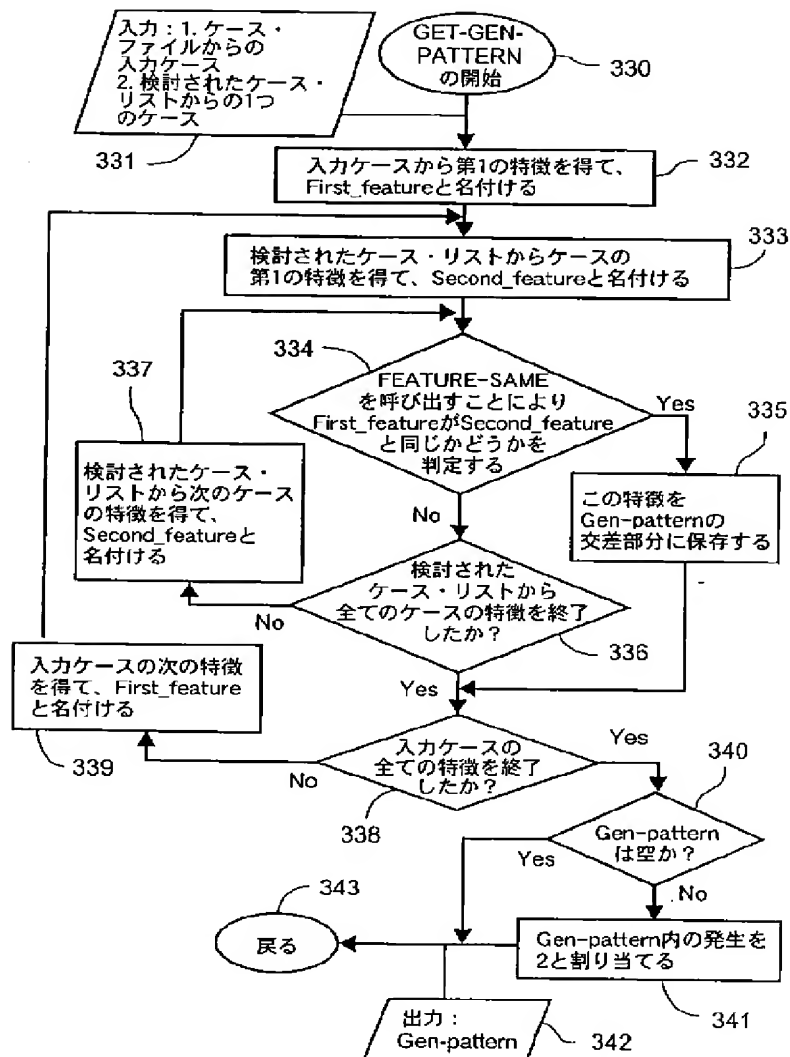
サブルーチン **FEATURE-SAME**

【図7H】



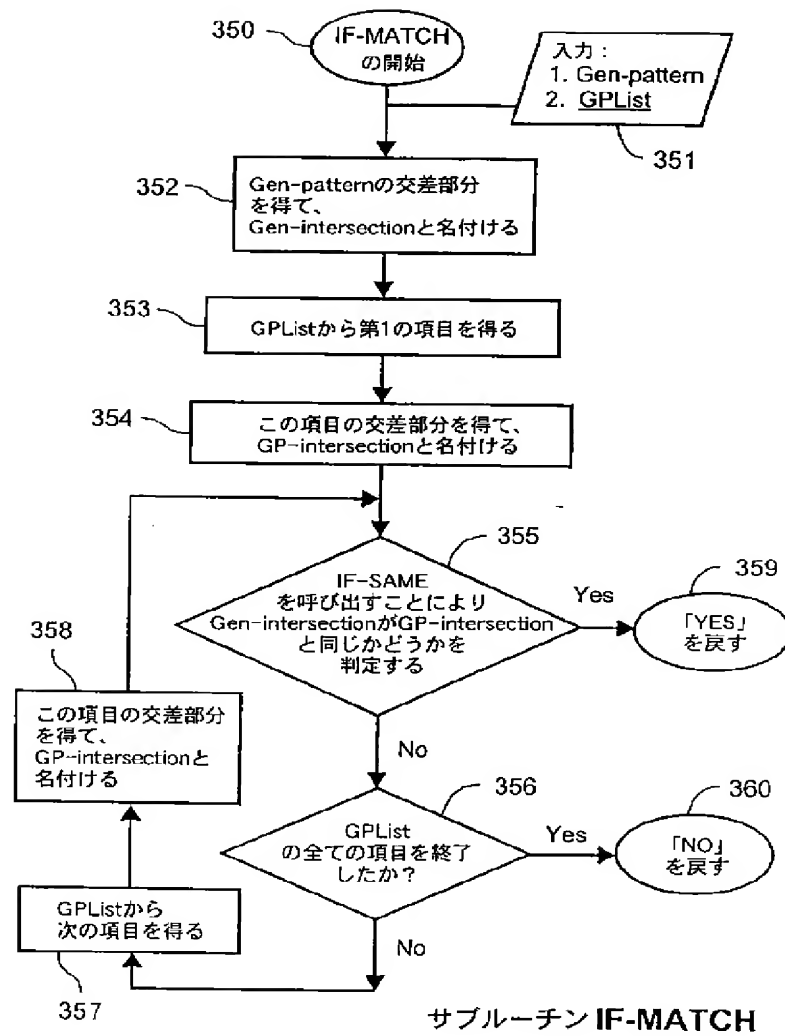
サブルーチン GEN-CASE-EXAMINED-CASES

【図7 I】

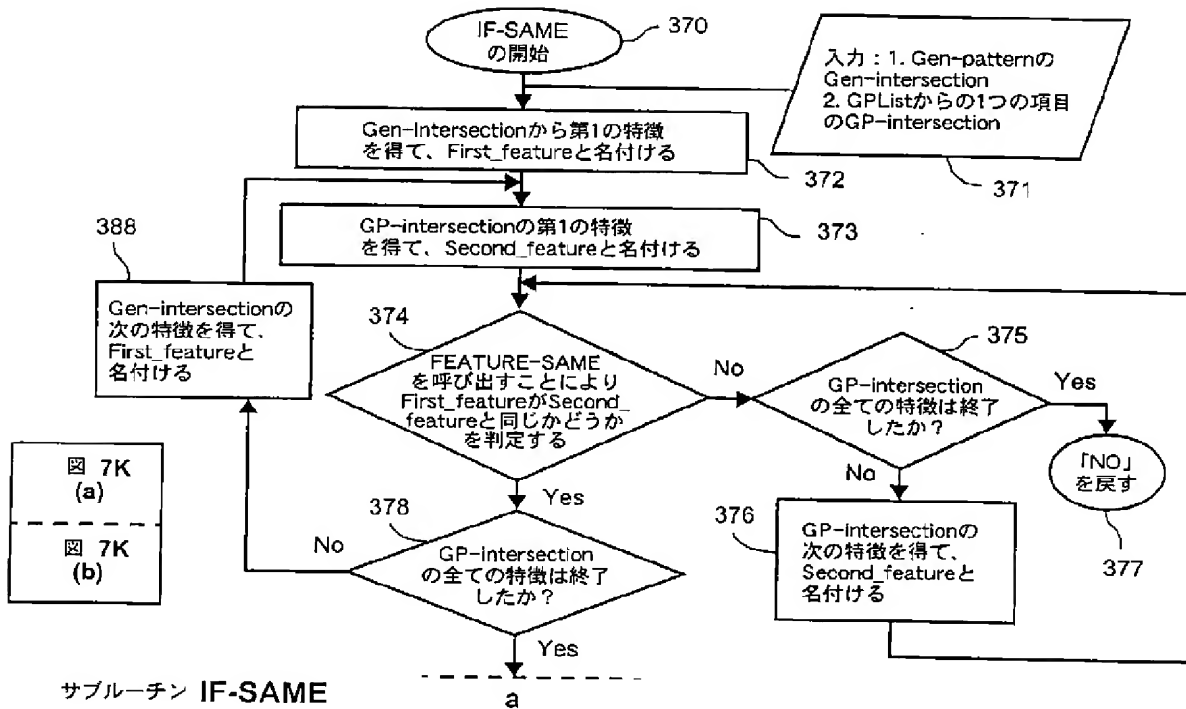


サブルーチン GET-GEN-PATTERN

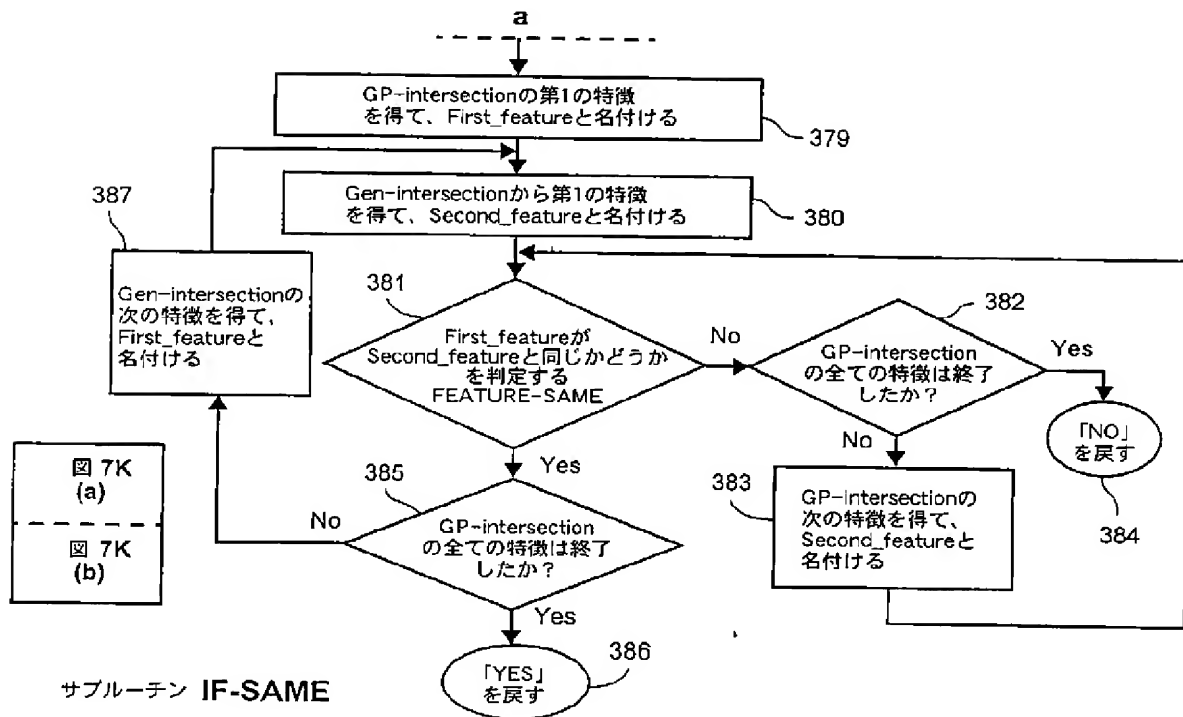
【図7J】



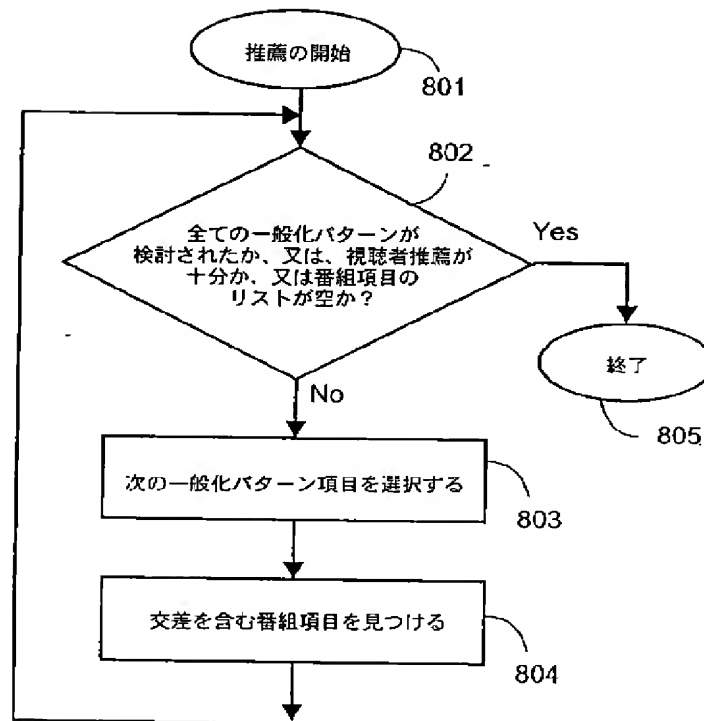
【図7K(a)】



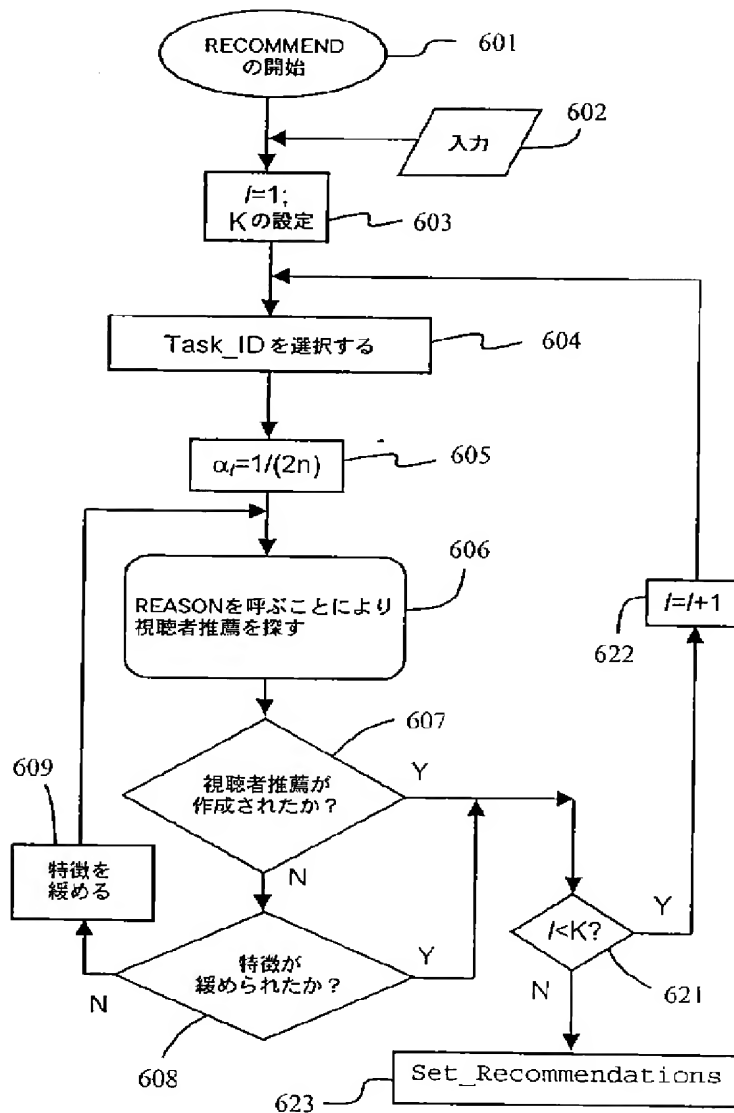
【図7K(b)】



【図8】

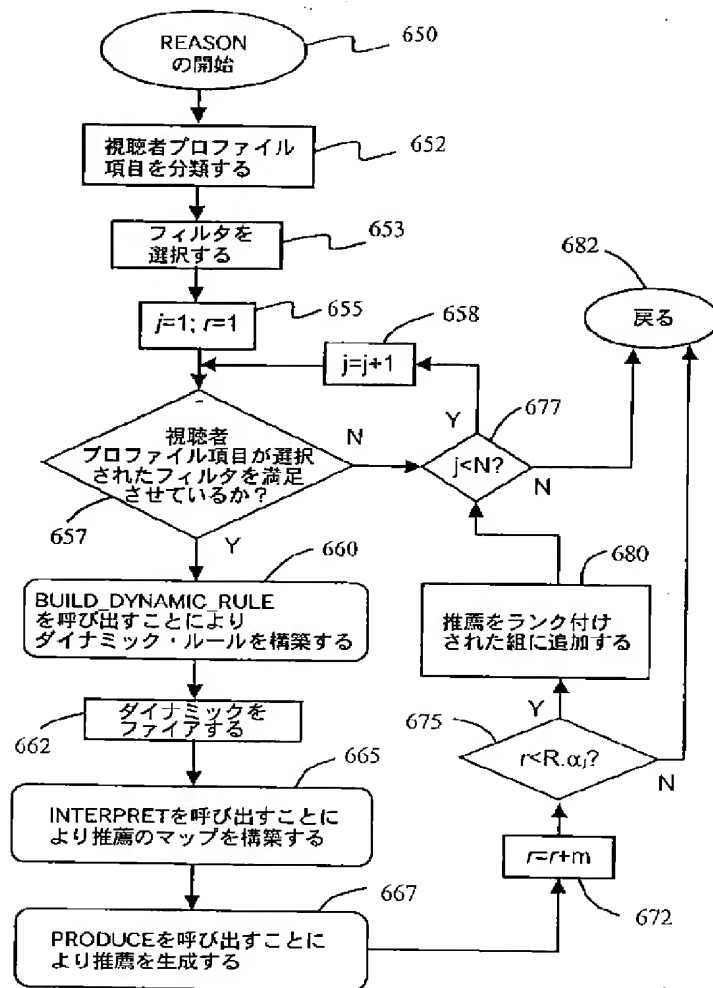


【図 9 A】



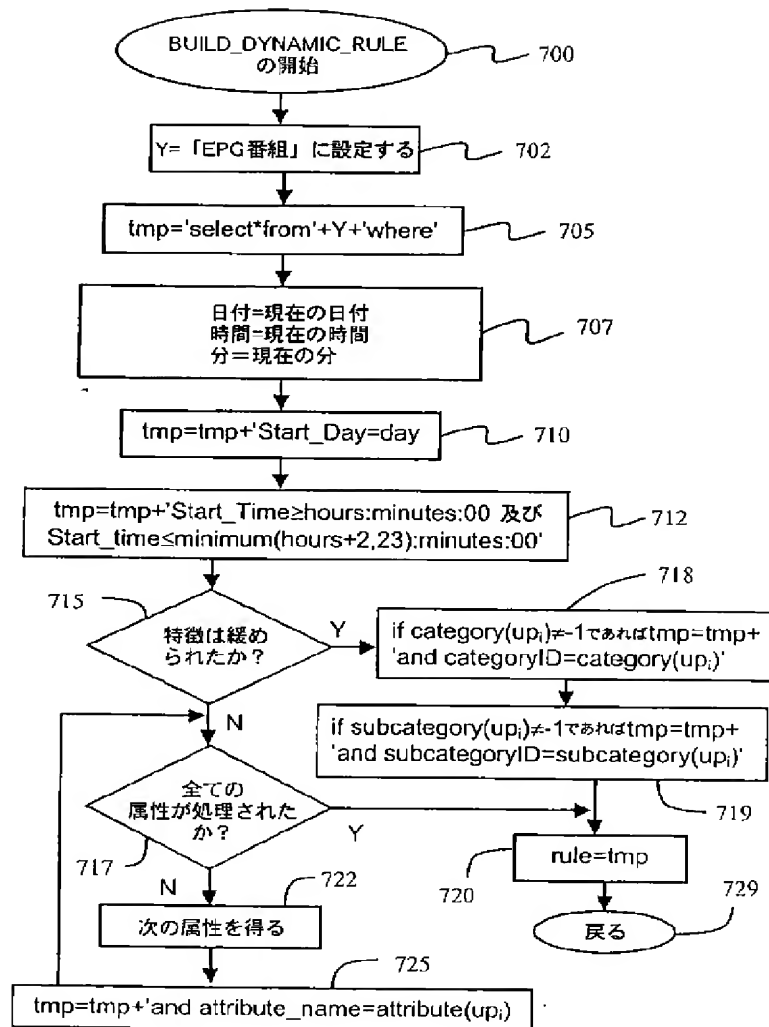
手順 RECOMMEND

【図 9 B】



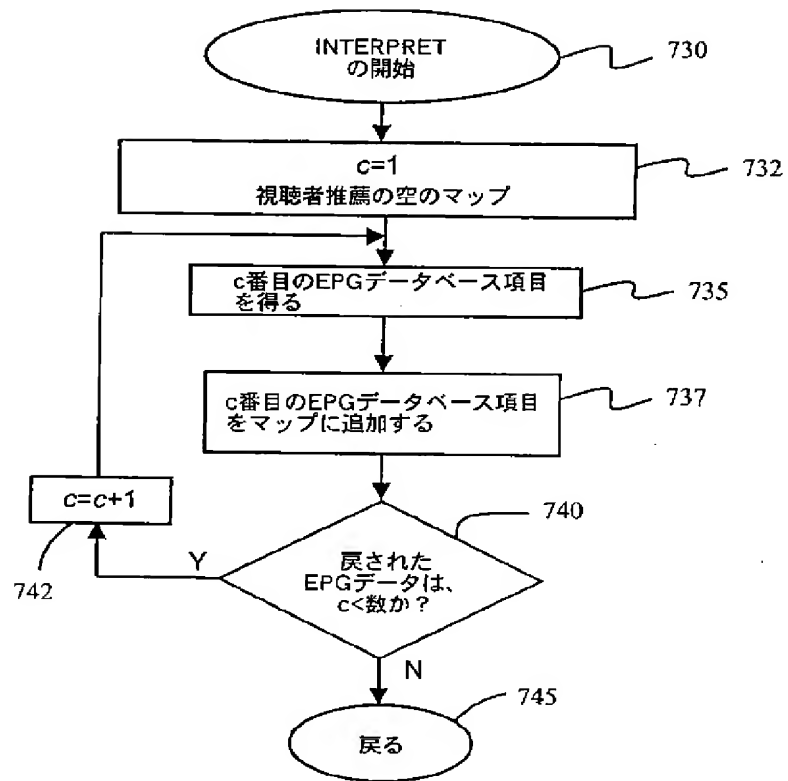
サブルーチン REASON

【図9C】



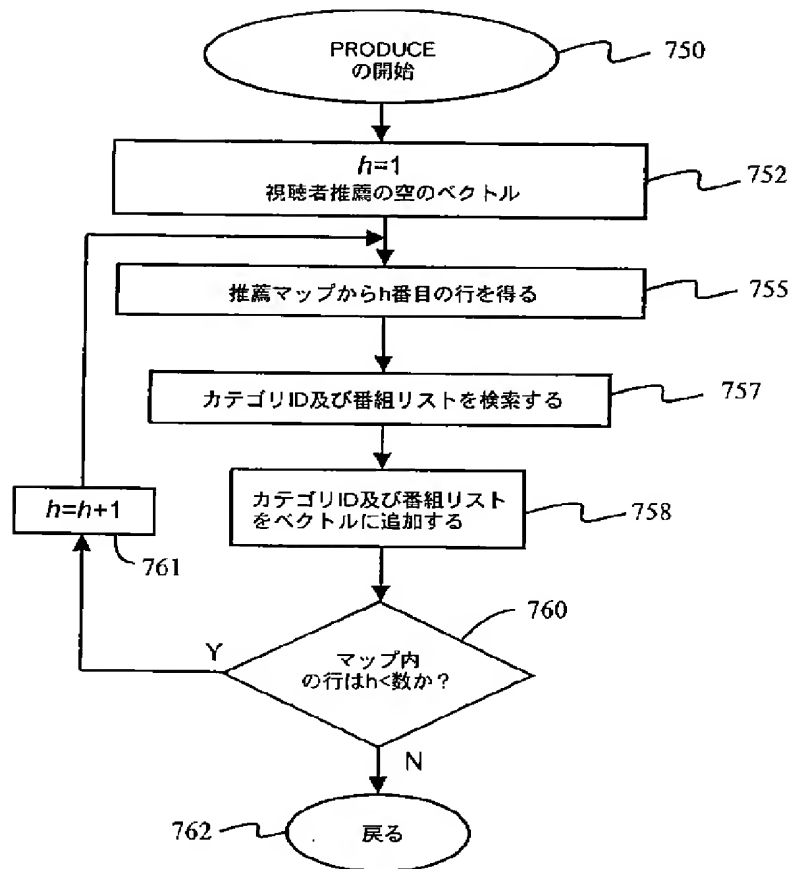
サブルーチン BUILD_DYNAMIC_RULE

【図9D】



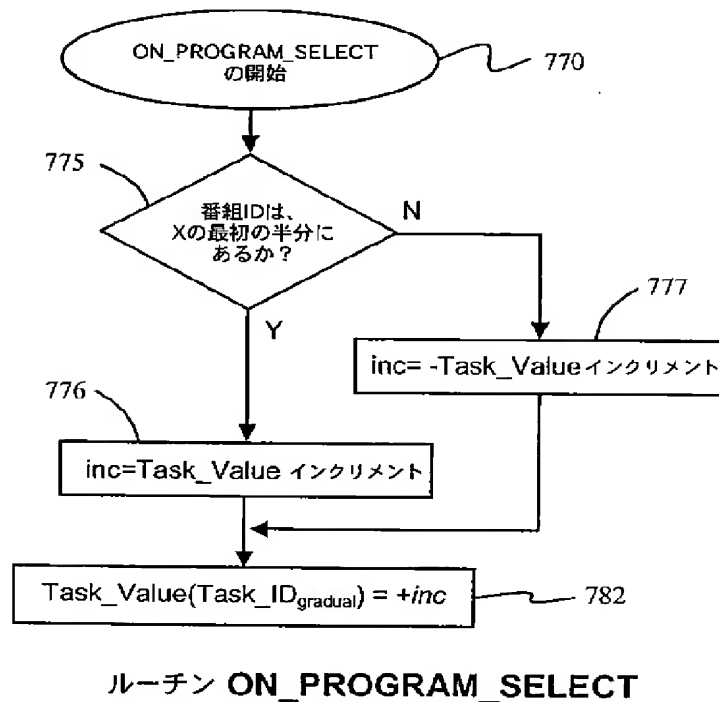
サブルーチン INTERPRET

【図9E】



サブルーチン PRODUCE

【図 9 F】



フロントページの続き

(51)Int.Cl.⁷

H 0 4 N 7/035

識別記号

F I

キーワード (参考)

(72)発明者 ワイ ヤット ウォン
オーストラリア国 2113 ニュー サウス
ウェールズ州、ノース ライド、トーマ
ス ホルト ドライブ 1 キヤノン イ
ンフォメーション システムズ リサーチ
オーストラリア プロプライエタリー
リミテッド 内

(72)発明者 ミハイル プロボベンコ
オーストラリア国 2113 ニュー サウス
ウェールズ州、ノース ライド、トーマ
ス ホルト ドライブ 1 キヤノン イ
ンフォメーション システムズ リサーチ
オーストラリア プロプライエタリー
リミテッド 内

(72)発明者 ファーハド ファド イスラム
オーストラリア国 2113 ニュー サウス
ウェールズ州、ノース ライド、トーマ
ス ホルト ドライブ 1 キヤノン イ
ンフォメーション システムズ リサーチ
オーストラリア プロプライエタリー
リミテッド 内

(72)発明者 リュザード ユヴァルツユク
オーストラリア国 2113 ニュー サウス
ウェールズ州、ノース ライド、トーマ
ス ホルト ドライブ 1 キヤノン イ
ンフォメーション システムズ リサーチ
オーストラリア プロプライエタリー
リミテッド 内

(72)発明者 マイケル アレキサンダー オールドフィ
ールド
オーストラリア国 2113 ニュー サウス
ウェールズ州, ノース ライド, トーマ
ス ホルト ドライブ 1 キヤノン イ
ンフォメーション システムズ リサーチ
オーストラリア プロプライエタリー
リミテッド 内

(72)発明者 マーク バルダー
オーストラリア国 2113 ニュー サウス
ウェールズ州, ノース ライド, トーマ
ス ホルト ドライブ 1 キヤノン イ
ンフォメーション システムズ リサーチ
オーストラリア プロプライエタリー
リミテッド 内

(72)発明者 ポール トレイヤー
オーストラリア国 2113 ニュー サウス
ウェールズ州, ノース ライド, トーマ
ス ホルト ドライブ 1 キヤノン イ
ンフォメーション システムズ リサーチ
オーストラリア プロプライエタリー
リミテッド 内

【 外国語明細書 】

1. Title of the Invention

A TELEVISION PROGRAM RECOMMENDATION SYSTEM

2. Claims

1. A method of enabling a selection of a program for viewing in a television system in which title information and characteristics of programs are made available as EPG (Electronic Program Guide) data including at least one program guide list, the method comprising the steps of:

recording a plurality of characteristics associated with each program viewed by said user in the television system;

forming sets of said characteristics, each said set comprising at least two of said characteristics; and

associating at least each set with an ordered value representative of user's desire to view a particular program;

wherein upon entry of a user request for a program recommendation, performing a search of the EPG data for programs with characteristics that best match said sets and notifying said user of an availability of programs that best match said sets as program recommendations.

2. A method according to claim 1, wherein

said search is performed for programs that best match said sets and the user's information determined in accordance with user's manual operations.

3. A method according to claim 2, wherein

said user's information is including a mood being experienced by said user.

4. A method according to claim 1, wherein

said program recommendations are based on the programs that best match said sets comprising a greatest number of said characteristics.

5. A method according to claim 1, wherein

said program recommendations are based on the programs that best match said sets associated with a lowest value as the ordered value.

6. A method according to claim 1, wherein

said program recommendations are based on the programs that best match said sets associated with a highest value as the ordered value and comprising the greatest number of said characteristics.

7. A method according to claim 1, wherein

said program recommendations are based on the programs that best match said sets associated with a lowest value as the ordered value and comprising the greatest number of said characteristics.

8. A method according to claim 1, wherein

said sets of said characteristics is formed in response to that at least two of programs viewed by said user have the same characteristics.

9. A recommendation system for enabling a selection of a program for viewing in a television system in which title information and characteristics of programs are made available as EPG (Electronic Program Guide) data including at least one program guide list, the recommendation system comprising:

memory means for recording a plurality of characteristics associated with each program viewed by said user in the television system;

processing means for forming sets of said characteristics, each said set comprising at least two of said characteristics and for associating each said set with an ordered value representative of said user's desire to view a particular program;

searching means for performing a search of the EPG data for programs with characteristics that best match said sets; and

on-screen display means for notifying said user of an availability of programs that best match said sets as program recommendations upon entry of a user request for program recommendations.

10. A recommendation system according to claim 9, wherein

said search is performed for programs that best match said sets and the user's information determined in accordance with user's manual operations.

11. A recommendation system according to claim 10, wherein

said user's information is including a mood being experienced by said user.

12. A recommendation system according to claim 9, wherein
said program recommendations are based on the programs that best match said
sets comprising a greatest number of said characteristics.

13. A recommendation system according to claim 9, wherein
said program recommendations are based on the programs that best match said
sets associated with a lowest value as the ordered value.

14. A recommendation system according to claim 9, wherein
said program recommendations are based on the programs that best match said
sets associated with a highest value as the ordered value and comprising the greatest
number of said characteristics.

15. A recommendation system according to claim 9, wherein
said program recommendations are based on the programs that best match said
sets associated with a lowest value as the ordered value and comprising the greatest
number of said characteristics.

16. A recommendation system according to claim 9, wherein
said sets of said characteristics is formed in response to that at least two of
programs viewed by said user have the same characteristics.

17. A computer program product, having a computer readable medium,
having a computer program recorded therein, for enabling a selection of a program for
viewing in a television system in which title information and characteristics of programs
are made available as EPG (Electronic Program Guide) data including at least one
program guide list, the computer program product comprising:

computer program code for recording a plurality of characteristics associated
with each program viewed by said user in the television system;

computer program code for forming sets of said characteristics, each said set
comprising at least two of said characteristics; and

computer program code for associating at least each set with an ordered value representative of user's desire to view a particular program;

wherein upon entry of a user request for a program recommendation, computer program code for performing a search of the EPG data for programs with characteristics that best match said sets and notifying said user of an availability of programs that best match said sets as program recommendations.

18. A computer program product according to claim 17, wherein said search is performed for programs that best match said sets and the user's information determined in accordance with user's manual operations.

19. A computer program product according to claim 18, wherein said user's information is including a mood being experienced by said user.

20. A computer program product according to claim 17, wherein said program recommendations are based on the programs that best match said sets comprising a greatest number of said characteristics.

21. A computer program product according to claim 17, wherein said program recommendations are based on the programs that best match said sets associated with a lowest value as the ordered value.

22. A computer program product according to claim 17, wherein said program recommendations are based on the programs that best match said sets associated with a highest value as the ordered value and comprising the greatest number of said characteristics.

23. A computer program product according to claim 17, wherein said program recommendations are based on the programs that best match said sets associated with a lowest value as the ordered value and comprising the greatest number of said characteristics.

24. A computer program product according to claim 17, wherein

said sets of said characteristics is formed in response to that at least two of programs viewed by said user have the same characteristics.

25. A method of making recommendations to a user of programs for viewing in a television system wherein:

information regarding programs is made available as EPG (Electronic Program Guide) data including at least one program guide list;

a plurality of characteristics associated with each program previously viewed by said user, and each of a plurality of relationships and corresponding measures of acceptance of said user, with respect to program recommendations made in response to previous program requests, have been recorded;

said method comprising the steps of:

forming sets of said characteristics;

selecting from said relationships at least one relationship, wherein at least one of said selected relationships comprises a highest one of said measures of acceptance;

ordering said sets of characteristics using said selected relationship to produce ordered sets of characteristics;

filtering said ordered sets of characteristics using said selected relationship to produce filtered sets of characteristics;

performing a search of the EPG data for programs with characteristics that best match said filtered sets of characteristics, giving preference to highest filtered sets of characteristics; and

notifying said user of an availability of programs that best match said filtered sets as program recommendations upon entry of a user request for program recommendations.

26. A method as claimed in claim 25, wherein each said set comprising a plurality of said characteristics.

27. A method as claimed in claim 25, comprising the further step of updating said measure of acceptance.

28. A method as claimed in claim 27, wherein each of said relationships comprises a combination of at least a first type relationship and a second type relationship,

said measure of said first type relationship being updated as a result of user selections made over an extended period of time and said measure of said second type relationship being updated as a result of user selections made over a shorter period of time.

29. A method as claimed in claim 28, wherein each of said relationships comprises a combination of said first type relationship having a maximum measure of acceptance and all second type relationships.

30. A method as claimed claim 25, wherein said relationships are user specific.

31. A method as claimed claim 25, wherein each of said sets of characteristics has associated therewith a frequency of occurrence of each said set in programs previously viewed by said user.

32. A method as claimed claim 25, wherein each of said sets of characteristics has associated therewith a date, said date being indicative of when each said set occurred in programs previously viewed by said user.

33. A method as claimed in claim 25 or 26, wherein each of said relationships are defined by a filtering and an ordering command.

34. A method as claimed in claim 33, wherein said filtering command is selected from the list including TimeSlot Filter, WeightFactor Filter, Popularity Filter, Urgency Filter and LifeStyle Filter as herein defined.

35. A method as claimed in claim 34, wherein:

said TimeSlot Filter depends on a day or time being suitable that for which recommendations are requested, said day or time being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user;

said WeightFactor Filter depends on a frequency of occurrence of each said set in programs previously viewed by said user,

said Popularity Filter depends on a popularity value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to a popular event/program.

said Urgency Filter depends on an urgency value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to an urgent program; and

said LifeStyle Filter depends on a lifestyle value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to a type of a lifestyle.

36. A method as claimed in claim 33, wherein said ordering command is selected from the list including WeightFactor Ordering, Recency Ordering, WeightRecency Ordering as herein defined.

37. A method as claimed in claim 36, wherein:

said WeightFactor Ordering arranges said sets of characteristics by descending a frequency of occurrence of each said set in programs previously viewed by said user;

said Recency Ordering arranges said sets of characteristics by descending a date being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user; and

said WeightRecency Ordering arranges said sets of characteristics by descending said frequency of occurrence, while preferring set of characteristics having most recent date when said frequency are the same.

38. A method as claimed in claim 26, wherein each of said relationships are defined by filtering and an ordering command, said ordering command being selected from the list including WeightFactor Ordering, Specificity Ordering, Generality Ordering, Recency Ordering, WeightRecency Ordering, Specificity Weight Ordering, Generality Weight Ordering, Recency Specificity Ordering, Specificity Recency Ordering and Generality Recency Ordering as herein defined.

39. A method as claimed in claim 38, wherein:

said WeightFactor Ordering arranges said sets of characteristics by descending a frequency of occurrence of each said set in programs previously viewed by said user;

said Specificity Ordering arranges said sets of characteristics by descending a specificity level corresponding to the number of said characteristics being comprised in said set;

said Generality Ordering arranges said sets of characteristics by ascending a specificity level;

said Recency Ordering arranges said sets of characteristics by descending a date being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user;

said WeightRecency Ordering arranges said sets of characteristics by descending said frequency of occurrence, while giving preference to set of characteristics having most recent data when said frequency are the same;

said Specificity Weight Ordering arranges said sets of characteristics by descending said specificity level, while giving preference to set of characteristics having higher frequency of occurrence when said specificity level are the same;

said Recency Specificity Ordering arranges said sets of characteristics by descending said date, while giving preference to set of characteristics having higher specificity level when said date are the same;

said Specificity Recency Ordering arranges said sets of characteristics by descending said specificity level, while giving preference to set of characteristics having most recent date when said specificity level are the same; and

said Generality Recency Ordering arranges said sets of characteristics by ascending said specificity level, while giving preference to set of characteristics having most recent date when said specificity level are the same.

40. A recommendation system for making recommendations to a user of programs for viewing in a television system in which information regarding programs is made available as EPG (Electronic Program Guide) data including at least one program guide list, the system comprising:

a first memory means for recording a plurality of characteristics associated with each program previously viewed by said user;

a second memory means for recording each of a plurality of relationships and corresponding measures of acceptance of said user, with respect to program recommendations made in response to previous program requests;

processing means for forming sets of said characteristics;

task selection means for selecting from said relationships at least one relationship, wherein at least one of said selected relationships comprises a highest one of said measures of acceptance;

ordering means for ordering said sets of characteristics using said selected relationship to produce ordered sets of characteristics;

filtering means for filtering said ordered sets of characteristics using said selected relationship to produce filtered sets of characteristics;

searching means performing a search of the EPG data for programs with characteristics that best match said filtered sets of characteristics, giving preference to highest filtered sets of characteristics; and

on-screen display means for notifying said user of an availability of programs that best match said filtered sets as program recommendations upon entry of a user request for program recommendations.

41. A recommendation system as claimed in claim 40, wherein each said set comprising a plurality of said characteristics.

42. A recommendation system as claimed in claim 40, further comprising a update means for updating said measures of acceptance.

43. A recommendation system as claimed in claim 42, wherein each of said relationships comprises a combination of at least a first type relationship and a second type relationship, said measure of said first type relationship being updated as a result of user selections made over an extended period of time and said measure of said second type relationship being updated as a result of user selections made over a shorter period of time.

44. A recommendation system as claimed in claim 43, wherein each of said relationships comprises a combination of said first type relationship having a maximum measure of acceptance and all second type relationships.

45. A recommendation system as claimed in claim 40, wherein said relationships are user specific.

46. A recommendation system as claimed in claim 40, wherein each of said sets of characteristics has associated therewith a frequency of occurrence of each said set in programs previously viewed by said user.

47. A recommendation system as claimed in claim 40, wherein each of said sets of characteristics has associated therewith a date, said date being indicative of when each said set occurred in programs previously viewed by said user.

48. A recommendation system as claimed in claim 40 or 41, wherein each of said relationships are defined by a filtering and a ordering command.

49. A recommendation system as claimed in claim 48, wherein said filtering command is selected from the list including TimeSlot Filter, WeightFactor Filter, Popularity Filter, Urgency Filter and LifeStyle Filter, as herein defined

50. A recommendation system as claimed in claim 49, wherein:

said TimeSlot Filter depends on a day or time being suitable that for which recommendations are requested, said day or time being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user;

said WeightFactor Filter depends on a frequency of occurrence of each said set in programs previously viewed by said user,

said Popularity Filter depends on a popularity value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to a popular event/program.

said Urgency Filter depends on an urgency value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to an urgent program; and

said LifeStyle Filter depends on a lifestyle value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to a type of a lifestyle.

51. A recommendation system as claimed in claim 48, wherein said ordering command is selected from the list including WeightFactor Ordering, Recency Ordering, WeightRecency Ordering as herein defined.

52. A recommendation system as claimed in claim 51, wherein:
said WeightFactor Ordering arranges said sets of characteristics by descending a frequency of occurrence of each said set in programs previously viewed by said user;
said Recency Ordering arranges said sets of characteristics by descending a date being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user; and
said WeightRecency Ordering arranges said sets of characteristics by descending said frequency of occurrence, while preferring set of characteristics having most recent date when said frequency are the same.

53. A method as claimed in claim 41, wherein each of said relationships are defined by filtering and an ordering command, said ordering command being selected from the list including WeightFactor Ordering, Specificity Ordering, Generality Ordering, Recency Ordering, WeightRecency Ordering, Specificity Weight Ordering, Generality Weight Ordering, Recency Specificity Ordering, Specificity Recency Ordering and Generality Recency Ordering as herein defined.

54. A recommendation system as claimed in claim 53, wherein:
said WeightFactor Ordering arranges said sets of characteristics by descending a frequency of occurrence of each said set in programs previously viewed by said user;
said Specificity Ordering arranges said sets of characteristics by descending a specificity level corresponding to the number of said characteristics being comprised in said set;
said Generality Ordering arranges said sets of characteristics by ascending a specificity level;
said Recency Ordering arranges said sets of characteristics by descending a date being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user;

said WeightRecency Ordering arranges said sets of characteristics by descending said frequency of occurrence, while giving preference to set of characteristics having most recent data when said frequency are the same;

said Specificity Weight Ordering arranges said sets of characteristics by descending said specificity level, while giving preference to set of characteristics having higher frequency of occurrence when said specificity level are the same;

said Recency Specificity Ordering arranges said sets of characteristics by descending said date, while giving preference to set of characteristics having higher specificity level when said date are the same;

said Specificity Recency Ordering arranges said sets of characteristics by descending said specificity level, while giving preference to set of characteristics having most recent date when said specificity level are the same; and

said Generality Recency Ordering arranges said sets of characteristics by ascending said specificity level, while giving preference to set of characteristics having most recent date when said specificity level are the same.

55. A computer program product, having a computer readable medium, having a computer program recorded therein, for making recommendations to a user of programs for viewing in a television system wherein:

information regarding programs is made available as EPG (Electronic Program Guide) data including at least one program guide list;

a plurality of characteristics associated with each program previously viewed by said user, and each of a plurality of relationships and corresponding measures of acceptance of said user, with respect to program recommendations made in response to previous program requests, have been recorded;

said computer program product comprising:

code for forming sets of said characteristics;

code for selecting from said relationships at least one relationship, wherein at least one of said selected relationships comprises a highest one of said measures of acceptance;

code for ordering said sets of characteristics using said selected relationship to produce ordered sets of characteristics;

code for filtering said ordered sets of characteristics using said selected relationship to produce filtered sets of characteristics;

code for performing a search of the EPG data for programs with characteristics that best match said filtered sets of characteristics, giving preference to highest filtered sets of characteristics; and

code for notifying said user of an availability of programs that best match said filtered sets as program recommendations upon entry of a user request for program recommendations.

56. A computer program product as claimed in claim 55, wherein each said set comprising a plurality of said characteristics.

57. A computer program product as claimed in claim 55, further comprising code for updating said measure of acceptance.

58. A computer program product as claimed in claim 57, wherein each of said relationships comprises a combination of at least a first type relationship and a second type relationship, said measure of said first type relationship being updated as a result of user selections made over an extended period of time and said measure of said second type relationship being updated as a result of user selections made over a shorter period of time.

59. A computer program product as claimed in claim 58, wherein each of said relationships comprises a combination of said first type relationship having a maximum measure of acceptance and all second type relationships.

60. A computer program product as claimed in claim 55, wherein said relationships are user specific.

61. A computer program product as claimed in claim 55, wherein each of said sets of characteristics has associated therewith a frequency of occurrence of each said set in programs previously viewed by said user.

62. A computer program product as claimed in claim 55, wherein each of said sets of characteristics has associated therewith a date, said date being indicative of when each said set occurred in programs previously viewed by said user.

63. A computer program product as claimed in any one of claims 55 or 56, wherein each of said relationships are defined by a filtering and a ordering command.

64. A computer program product as claimed in claim 63, wherein said filtering command is selected from the list including TimeSlot Filter, WeightFactor Filter, Popularity Filter, Urgency Filter and LifeStyle Filter as herein defined.

65. A computer program product as claimed in claim 64, wherein:

said TimeSlot Filter depends on a day or time being suitable that for which recommendations are requested, said day or time being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user;

said WeightFactor Filter depends on a frequency of occurrence of each said set in programs previously viewed by said user,

said Popularity Filter depends on a popularity value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to a popular event/program.

said Urgency Filter depends on an urgency value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to an urgent program; and

said LifeStyle Filter depends on a lifestyle value being comprised in said sets of characteristics and indicative that each program previously viewed by said user corresponds to a type of a lifestyle.

66. A computer program product as claimed in claim 63, wherein said ordering command is selected from the list including WeightFactor Ordering, Recency Ordering, WeightRecency Ordering as herein defined.

67. A computer program product as claimed in claim 66, wherein:

said WeightFactor Ordering arranges said sets of characteristics by descending a frequency of occurrence of each said set in programs previously viewed by said user;

said Recency Ordering arranges said sets of characteristics by descending a date being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user; and

said WeightRecency Ordering arranges said sets of characteristics by descending said frequency of occurrence, while preferring set of characteristics having most recent date when said frequency are the same.

68. A computer program product as claimed in claim 66, wherein each of said relationships are defined by filtering and an ordering command, said ordering command being selected from the list including WeightFactor Ordering, Specificity Ordering, Generality Ordering, Recency Ordering, WeightRecency Ordering, Specificity Weight Ordering, Generality Weight Ordering, Recency Specificity Ordering, Specificity Recency Ordering and Generality Recency Ordering as herein defined.

69. A computer program product as claimed in claim 68, wherein:

said WeightFactor Ordering arranges said sets of characteristics by descending a frequency of occurrence of each said set in programs previously viewed by said user;

said Specificity Ordering arranges said sets of characteristics by descending a specificity level corresponding to the number of said characteristics being comprised in said set;

said Generality Ordering arranges said sets of characteristics by ascending a specificity level;

said Recency Ordering arranges said sets of characteristics by descending a date being comprised in said sets of characteristics and indicative of when each said set occurred in programs previously viewed by said user;

said WeightRecency Ordering arranges said sets of characteristics by descending said frequency of occurrence, while giving preference to set of characteristics having most recent data when said frequency are the same;

said Specificity Weight Ordering arranges said sets of characteristics by descending said specificity level, while giving preference to set of characteristics having higher frequency of occurrence when said specificity level are the same;

said Recency Specificity Ordering arranges said sets of characteristics by descending said date, while giving preference to set of characteristics having higher specificity level when said date are the same;

said Specificity Recency Ordering arranges said sets of characteristics by descending said specificity level, while giving preference to set of characteristics having most recent date when said specificity level are the same; and

said Generality Recency Ordering arranges said sets of characteristics by ascending said specificity level, while giving preference to set of characteristics having most recent date when said specificity level are the same.

3. Detailed Description of the Invention

Field of the Invention

The present invention relates to television program selection and more particularly to a system for providing recommendations for television program selection.

Description of Background Art

Watching television is an everyday activity in a large number of households, and provides for a source of entertainment for a range of program content such as sport and movies, as well as news and actuality programs (eg. documentaries and lifestyle).

Traditionally, a viewer would consult a printed television schedule listing in a local newspaper or periodical to find a desirable program to watch. More recently, entertainment program guides have been made available in electronic form via the Internet or World Wide Web. However, with the introduction of satellite receivers for television and cable television, the number of television channels available to the viewer has increased dramatically. This has made the task of selecting a television program to watch using either printed program listings or on-screen Electronic Program Guides (EPG) very involved and time consuming.

Program listings that are ordered by characteristics including category, time or performers, can assist the viewer in making a selection, but such remains a time consuming task. Typically, there may only be a few programs of interest to the viewer out of the vast number of available programs.

Systems are available that monitor those categories of programs a viewer most frequently watches and to provide recommendations based on the most frequently watched category. Such arrangements suffer from the disadvantage of simply selecting a single program characteristic (such as category) from previously watched programs and making recommendations based on this selected program characteristic. However, selection by the viewer may depend on various characteristics and even combinations of characteristics. Furthermore, exactly how these characteristics and combinations of

characteristics are joined for use in making the selection is highly individual for each user and prior art systems do not provide this level of flexibility.

Disclosure of the Invention

It is an object of the present invention to substantially overcome, or at least ameliorate, one or more disadvantages of existing arrangements.

According to a first aspect of the invention, there is provided a method of enabling a selection of a program for viewing in a television system in which title information and characteristics of programs are made available as EPG (Electronic Program Guide) data including at least one program guide list, the method comprising the steps of:

recording a plurality of characteristics associated with each program viewed by said user in the television system;

forming sets of said characteristics, each said set comprising at least two of said characteristics; and

associating at least each set with an ordered value representative of user's desire to view a particular program;

wherein upon entry of a user request for a program recommendation, performing a search of the EPG data for programs with characteristics that best match said sets and notifying said user of an availability of programs that best match said sets as program recommendations.

According to another aspect of the invention there is provided a recommendation system for enabling a selection of a program for viewing in a television system in which title information and characteristics of programs are made available as EPG (Electronic Program Guide) data including at least one program guide list, the recommendation system comprising:

memory means for recording a plurality of characteristics associated with each program viewed by said user in the television system;

processing means for forming sets of said characteristics, each said set comprising at least two of said characteristics and for associating each said set with an ordered value representative of said user's desire to view a particular program;

searching means for performing a search of the EPG data for programs with characteristics that best match said sets; and

on-screen display means for notifying said user of an availability of programs that best match said sets as program recommendations upon entry of a user request for program recommendations.

According to yet another aspect of the invention there is provided a computer program product, having a computer readable medium, having a computer program recorded therein, for enabling a selection of a program for viewing in a television system in which title information and characteristics of programs are made available as EPG (Electronic Program Guide) data including at least one program guide list, the computer program product comprising:

- computer program code for recording a plurality of characteristics associated with each program viewed by said user in the television system;

- computer program code for forming sets of said characteristics, each said set comprising at least two of said characteristics; and

- computer program code for associating at least each set with an ordered value representative of user's desire to view a particular program;

wherein upon entry of a user request for a program recommendation, computer program code for performing a search of the EPG data for programs with characteristics that best match said sets and notifying said user of an availability of programs that best match said sets as program recommendations.

According to yet another aspect of the invention there is provided a method of making recommendations to a user of programs for viewing in a television system wherein:

- information regarding programs is made available as EPG (Electronic Program Guide) data including at least one program guide list;

- a plurality of characteristics associated with each program previously viewed by said user, and each of a plurality of relationships and corresponding measures of acceptance of said user, with respect to program recommendations made in response to previous program requests, have been recorded;

- said method comprising the steps of:

- forming sets of said characteristics;

- selecting from said relationships at least one relationship, wherein at least one of said selected relationships comprises a highest one of said measures of acceptance;

- ordering said sets of characteristics using said selected relationship to produce ordered sets of characteristics;

filtering said ordered sets of characteristics using said selected relationship to produce filtered sets of characteristics;

performing a search of the EPG data for programs with characteristics that best match said filtered sets of characteristics, giving preference to highest filtered sets of characteristics; and

notifying said user of an availability of programs that best match said filtered sets as program recommendations upon entry of a user request for program recommendations.

According to yet another aspect of the invention there is provided a recommendation system for making recommendations to a user of programs for viewing in a television system in which information regarding programs is made available as EPG (Electronic Program Guide) data including at least one program guide list, the system comprising:

a first memory means for recording a plurality of characteristics associated with each program previously viewed by said user;

a second memory means for recording each of a plurality of relationships and corresponding measures of acceptance of said user, with respect to program recommendations made in response to previous program requests;

processing means for forming sets of said characteristics;

task selection means for selecting from said relationships at least one relationship, wherein at least one of said selected relationships comprises a highest one of said measures of acceptance;

ordering means for ordering said sets of characteristics using said selected relationship to produce ordered sets of characteristics;

filtering means for filtering said ordered sets of characteristics using said selected relationship to produce filtered sets of characteristics;

searching means performing a search of the EPG data for programs with characteristics that best match said filtered sets of characteristics, giving preference to highest filtered sets of characteristics; and

on-screen display means for notifying said user of an availability of programs that best match said filtered sets as program recommendations upon entry of a user request for program recommendations.

According to yet another aspect of the invention there is provided a computer program product, having a computer readable medium, having a computer program

recorded therein, for making recommendations to a user of programs for viewing in a television system wherein:

information regarding programs is made available as EPG (Electronic Program Guide) data including at least one program guide list;

a plurality of characteristics associated with each program previously viewed by said user, and each of a plurality of relationships and corresponding measures of acceptance of said user, with respect to program recommendations made in response to previous program requests, have been recorded;

said computer program product comprising:

code for forming sets of said characteristics;

code for selecting from said relationships at least one relationship, wherein at least one of said selected relationships comprises a highest one of said measures of acceptance;

code for ordering said sets of characteristics using said selected relationship to produce ordered sets of characteristics;

code for filtering said ordered sets of characteristics using said selected relationship to produce filtered sets of characteristics;

code for performing a search of the EPG data for programs with characteristics that best match said filtered sets of characteristics, giving preference to highest filtered sets of characteristics; and

code for notifying said user of an availability of programs that best match said filtered sets as program recommendations upon entry of a user request for program recommendations.

Detailed Description including Best Mode

Fig. 1A shows a system 50 for automatically suggesting suitable programs to a viewer from a large number of available programs. The system 50 comprises a digital television (DTV) 10 connected through interconnection 25 to a "set top" box 20. The "set top" box 20 is also connected through a gateway 41 to an external network 42. A number of content servers 43 and Electronic Program Guide (EPG) databases 22 are connected to the external network 42. The content servers 43 are typically provided by content providers and contain multimedia content including movies and television programs.

The available programs are listed in the EPG databases 22. Each of the content providers may maintain an associated EPG database 22. The available programs listed in

the EPG databases 22 may be linked to the corresponding multimedia content in the content servers 43 by means of a program identifier. An extract from a typical EPG database 22 is shown in Fig. 2. The EPG database 22 has a number of program entries 60, each of which may include a number of attributes 61 with values 62. The attributes 61 may include:

- EPG ID which is a unique number per entry;
- Program ID which is unique for each program resulting in re-runs having the same Program ID;
- Category ID, for example '01' for art, '16' for drama, '50' for Movies and Series, '75' for sport etc.;
- Subcategory ID, for example, in the movies category '50', the subcategory '001' for action/adventure, '064' for comedy, '074' for crime etc;
- Title;
- Remarks, which is a general text field about the program entry;
- Keywords;
- Rating;
- EPG Channel;
- Start Day;
- Start Date;
- Start time;
- Duration; and
- Year of Make.

A DTV-agent system 21 is preferably formed within the "set top" box 20. In use, the viewer interacts with the DTV-agent system 21 using a remote control device 30. The DTV-agent system 21 may alternatively be integrated into the DTV 10 or incorporated into a personal computer 100 such as that seen in Fig. 10 and appropriately interfaced with the DTV 10.

In the preferred implementation, the DTV-agent system 21 includes a DTV-agent 36 for controlling the DTV 10, and a number of avatar-agents 37, each avatar-agent 37 representing a particular viewer. An Inter-agent-server (IAS) 35 manages all inter-agent communications and also interfaces with the gateway 41. The DTV-agent system 21 further includes a number of viewer profile databases 23 accessed by the avatar-agents 37. The functions of the DTV-agent 36 include:

- the provision of a graphical user interface via a display screen 11 of the DTV 10 by which recommendations are made to the viewer and to enable the viewer to make selections;
- to control the functionality of the DTV 10 including interacting with a television tuner module (not illustrated) thereof to select a channel for viewing from the plurality of available television channels; and
- to gather viewer selections and delivers them to the avatar agents 37.

Each of the avatar-agents 37 is associated with a single viewer and a corresponding one of the viewer profile databases 23. As seen in Fig. 1B, each avatar-agent 37 includes an avatar manager 38 that maintains control of the particular avatar-agent 37. The avatar manager 38 accesses the EPG databases 22 through the IAS 35 to obtain television program information and the associated viewer profile database 23 to build and maintain the viewer profile. The avatar manager 38 is also responsible for sending messages to, and receiving messages from, the DTV-agent 36 through the IAS 35. Within the avatar-agent 37, the avatar manager 38 is also responsible for interfacing with each of a learning module 39 and a recommendation module 40. One task of the avatar-agent 37 is to build and maintain the viewer profile database 23 based on selections made by the viewer and processed by the learning module 39. The avatar-agent 37 then also uses the viewer profile database 23 to provide recommendations to the viewer by processing performed in the recommendation module 40. The recommendations are for the most suitable viewing options available for a specific viewer and are provided to the viewer via the display screen 11.

Referring again to Fig 1A, the system 50 may further include an EPG agent 44 that interfaces with both the EPG databases 22 and the IAS 35. The EPG agent 44 accesses the EPG databases 22 and extract further attributes from the program entries 60, which includes Popularity, Urgency and Lifestyle. The Popularity attribute indicates whether a program entry 60 corresponds to a (currently) popular event/program, such as the Football World Cup or a concert by a popular performer. The Urgency attribute indicates whether a program entry 60 corresponds with an urgent program, such as special news/progress reports, weather announcements (tornado warnings, etc.), press conferences, nation addresses and so on. The Lifestyle attribute indicates that the program entry 60 corresponds to a certain type of lifestyle, such as Midday movies, 7pm news, or Saturday night music shows. The values of these further attributes are True or

False. In an alternative implementation, these attributes are already included in the EPG databases 22.

Referring to Tables 1 and 2, examples of typical steps for use of the system 50 will now be described.

TABLE 1

Step 1#	Message	From	To	Internal processing
2				Viewer switches DTV 10 on. Animation characters 12 are displayed on the display screen 11. Viewer chooses animation character 12.
3	SetAnimationSelection	DTVA	AA	
4#	SendUserData	AA	DTVA	
5				Viewer interactively provides personal information and a viewing history based on prompts displayed on the display screen 11.
6	SetNewUserData	DTVA	AA	
*				
7#				Viewer asks for recommendations 67.
8				Prompt viewer for time and date.
9	SendRecommendationRequest	DTVA	AA	
*				
10#	SetRecommendations	AA	DTVA	
11				EPG and preview data retrieved.
12				Preview data are displayed on display screen 11. Viewer makes a selection 68.
13	SetProgramSelection	DTVA	AA	
*				

Note: When the sequence shown in Table 1 reaches a star symbol (*), the DTV-agent 36 enters an "idle" mode, waiting for messages from avatar agent 37. In this mode, the DTV 10 displays a television program (on a channel most recently selected by the viewer). Sequence points indicated by a hash symbol (#) can be accessed at any time (i.e. asynchronously).

Table 1 illustrates a sequence of events and resulting messages to and from the DTV-agent 36 in which:

1. The user switches the DTV 10 'ON' using the remote control 30.
2. The viewer needs to identify himself/herself to the DTV-agent system 21.

Referring to Fig. 3, this may be done by selecting an animation character 12

from a set of characters generated by the DTV-agent 36 and displayed on the display screen 11. The selection may be performed using the remote control 30.

In the preferred implementation, there may be three types of characters:

- (i) Un-touched characters, meaning those that are not associated with any viewer;
- (ii) Dormant characters, meaning those that are associated with a viewer but that particular viewer is currently not engaged; and
- (iii) An active character 12, identifying that the associated viewer is presently selected. No active character will be present when the DTV 10 has just been turned 'ON'.

3. A SetAnimationSelection message is sent to the avatar agents 37, activating only the avatar-agent 37 associated with the identified viewer.
4. A SendUserData message is received from the avatar agent 37, instructing the DTV agent 36 to collect viewer data.
5. The DTV-agent 36 controls the display screen 11 of the DTV 10 for the collection of viewer data. The viewer provides the data using the remote control 30. The first time that a character 12 is selected, the system 50 assumes that it is a new viewer using the system 50 and therefore an unregistered viewer. The avatar agent 37 has no viewer profile database for the currently unregistered viewer.

The viewer may then be requested by the DTV-agent 36 to select one of a number of moods currently being experienced, again by making a selection with the remote control 30. Examples of such moods include happy, sad, neutral etc. A password may also be requested from the viewer when the program content is password-protected (eg. adult content).

6. The viewer data are formatted and sent in a SetNewUserData message to the avatar-agent 37. The DTV agent 36 enters into the idle mode.
7. The viewer uses the remote control 30 to request from the DTV-agent system 21 a list of recommended television programs.
8. The DTV-agent 36 prompts the viewer for a time and date for which recommendations should be provided. Alternatively, recommendations may automatically be for television programs due to start in the following predetermined number of minutes.

9. The time and date are formatted and sent to the avatar-agent 37 in a SendRecommendationRequest message. The DTV agent 36 enters into the idle mode.
10. The avatar agent 37, and in particular the recommendation module 40, determines viewing recommendations for viewing from the EPG databases 22 that best match the viewer's preferences for the given timeslot and the viewer mood. The avatar agent 37 sends these viewing recommendations in a SetRecommendations message to the DTV agent 36.
11. Based on the recommended programs received from the avatar agent 37, data are retrieved from the EPG databases 22 and one or more preview databases (not shown). The preview databases may form part of the EPG databases 22 or alternatively, the content servers 43 and provide a brief example or preview of the content of a particular program.
12. The DTV-agent 36 formats the viewing recommendations and displays the recommendations on the display screen 11. As illustrated in Fig. 4, the viewing recommendations 67 are preferably provided to the viewer on the display screen 11 by displaying a set of previews. Alternatively, a bitmap image or a title only is displayed on the display screen 11, each of which representing a viewing recommendation 67. Additional information may also be displayed associated with each of the recommendations 67. The viewing recommendations 67 may also be displayed in a manner to reflect the program attributes. For example, all the viewing recommendations 67 having a common category may be displayed together. The viewer views the viewing recommendations 67 and selects one of the viewing recommendations 67 using the remote control 30. The DTV-agent 36 interacts with the DTV 10, which responds by tuning into a receiver channel on which the selected program is broadcast. The selected program 68 is then displayed on the full display screen 11 as shown in Fig. 5.
13. Information about the selected program 68 is sent to the avatar agent 37 in a SetProgramSelection message. The DTV-agent 36 enters its idle mode while the selected program 68 is displayed on the display screen 11.

Table 2 illustrates a sequence of events and resulting messages to and from the avatar-agents 37 in which:

1. A SetAnimationSelection message is received from the DTV-agent 36 by the avatar-agent 37 associated with the selected animation character 12. The other avatar-agents 37 remain in the idle mode.
2. The avatar agent 37 determines whether the selected character 12 corresponds to a viewer known to the DTV-agent system 21, or to a new viewer. If the viewer is not known to the DTV-agent system 21, the next sequence step is step 3. Otherwise the avatar manager 38 enters its idle mode.

TABLE 2

Step 1#	Message	From DTVA	To AA	Internal processing
2	SetAnimationSelection			Retrieve information from viewer profile database 23 to establish whether the viewer has profile data. If not, go to step 3.
*				
3				Create viewer profile database 23.
4	SendViewerData	AA	DTVA	
5	SetNewViewerData	DTVA	AA	
6				Update viewer profile database 23.
*				
7#	SendRecommendationRequest	DTVA	AA	
8				Retrieve viewer profile and EPG information from databases 23 and 22.
9				Send viewer data and television program information to recommendation module 40.
10				Recommendation module 40 constructs a list of recommended television programs 67.
11				Receive program recommendations 67 from recommendation module 40.
12	SetRecommendations	AA	DTVA	
*				
13#	SetProgramSelection	DTVA	AA	
14				Retrieve viewer profile and program data from databases 23 and 22.
15				Send viewer data and new program data to learning module 39.
16				The learning module 39 updates the Case-file.
17				Receive Case-file from learning module 39.
18				Update viewer profile database 23.
*				

Note: When the sequence shown in Table 1 reaches a *, the avatar-agent 37 enters an "idle" mode, waiting for messages from the DTV-agent 36. In this mode, background processing by the learning module 39 and the recommendation module 40 may occur. Sequence points indicated by a hash symbol (#) can be accessed at any time (i.e. asynchronously).

3. A profile for the new viewer is created in the viewer profile database 23.
4. A SendViewerData message is sent to the DTV agent 36.
5. A SetNewViewerData message is received from the DTV agent 36. This contains static viewer attributes.
6. The avatar manager 38 stores the viewer attributes in the viewer profile database 23. The avatar manager 38 enters its idle mode.
7. A SendRecommendationRequest message is received from the DTV agent 36. The viewer has requested that viewer recommendations 67 for a particular day and time be provided. Alternatively, recommendations may automatically be for television programs due to start in the following predetermined number of minutes.
8. The avatar-manager 38 retrieve viewer profile data from viewer profile database 23 and retrieve television program information from the EPG databases 22. The data is formatted for use by recommendation module 40.
9. The viewer profile data and the program information are sent to the recommendation module 40 by the avatar manager 38. Alternatively, the recommendation module 40 retrieves the data directly.
10. The recommendation module 40 constructs a list of programs 67 to be recommended to the viewer.
11. The list of recommended programs 67 is sent from the recommendation module 40 to the avatar manager 38.
12. A SetRecommendations message, containing a list of recommended programs 67 is sent to the DTV agent 36. The avatar manager 38 enters its idle mode.
13. A SetProgramSelection is received from the DTV agent 36, indicating that the viewer has selected a particular program 68.
14. The program entry 60 about the selected program 68 is retrieved from the EPG database 22. The viewer profile is also retrieved from the viewer profile database 23.
15. The selected program information is sent to the learning module 39 for updating of a Case-file.
16. The learning module 39 updates the Case-file to take account of the selected program 68.
17. The updated Case-file is received by the avatar manager 38 from the learning module 39.
18. The updated Case-file is used to update the viewer profile database 23. The avatar manager 38 enters its idle mode.

The operations within the learning module 39 will now be described in more detail. Each program is associated with a number of features (fi), each feature representing a unique attribute and attribute value pair. The attributes may include the program attributes 61 from the EPG database 22 (illustrated in Fig. 2) as well as viewer attributes when the selection was made. The viewer attributes may include for example the mood of the viewer. Each attribute has a number of possible values. An example of an attribute-value pair is *Category* = 'drama' where *Category* is the attribute and 'drama' is the attribute value.

Each selection 68 made by the viewer is termed a case (Cj), and is defined by a set of features (fi). For example, a case may be (*Category* = 'drama', *Sub-category* = 'comedy-drama', ..., *EPG channel* = 'Nhk', ...). The cases (Cj) may have different number of features (fi), dependent on the information available. All cases are stored in the viewer profile database 23 in a case file for use by the learning module 39.

The learning module 39 operates to identify shared or generalisation patterns of features (fi) from a number of cases (Cj) from the case file. The learning module 39 takes the case file as input and generates a Generalisation Pattern List (GPList), stored also in the viewer profile database 23. The GPList contains all generalisation patterns from the case file. Each generalisation pattern in the GPList may be represented as follows:

([intersection], occurrence),

wherein intersection indicates a set of the features (fi) that are shared by different cases (Cj). Occurrence indicates the number of cases that share such an intersection. For example, from the following cases:

C1 = (f1, f2, f3, f4, f7);

C2 = (f1, f2, f5, f6, f7); and

C3 = (f3, f5, f6, f8)

the GPList generated from the three cases has three items:

([f1, f2, f7], 2);

([f3], 2); and

([f5, f6], 2)

The first GPList entry above arises from the features f1, f2 and f7 occurring both in cases C1 and C2. The other entries are derived in a similar manner.

Each time new cases (Cj) become available through selections by the viewer of programs to watch, those new cases are added to the case file and a new GPList is

generated by the learning module 39. Assume for example that the case file contains the following three cases:

C1 = (Category = 'drama', Sub-category = 'comedy-drama', EPG channel = 'Nhk');

C2 = (Category = 'drama', Sub-category = 'war', EPG channel = 'Nhk', StartTime = '21:00'); and

C3 = (Category = 'sports', EPG channel = 'Wow', StartTime = '8:15', Day = 'Sat').

The GPList generated by the learning module would only include:

([Category = 'drama', EPG channel = 'Nhk'], 2)

When new viewer selections 68, and therefore further cases, became available, for example

C4 = (Category = 'drama', Sub-category = 'Talk', EPG channel = 'Nhk'); and

C5 = (Category = 'sports', Sub-category = 'Current affairs', StartTime = '10:00', Day = 'sat')

cases C4 and C5 are added to the case file and the GPList is recalculated by the learning module 39 to contain the following entries:

([Category = 'drama', EPG channel = 'Nhk'], 3)

([Category = 'sports', Day = 'Sat'], 2)

A procedure **MAIN** for performing the core function in the learning module 39 is shown in Fig. 7A. Three data structures are used in the procedure namely, the Case-file which contains a number of cases (Cj), the GPList which keeps all generalisation patterns, and an Examined-case-list which contains cases that are already processed. Initially, upon initiation of the procedure **MAIN** when a viewer requested for recommendations, the GPList and the Examined-case-list are empty.

The procedure **MAIN** starts in step 200 by first obtaining a list of new cases in step 201 and the Case-file from the viewer profile database in step 202. A sub-routine **UPDATE-CASE-FILE** is called in step 203 for updating the Case-file by adding the new cases. The sub-routine **UPDATE-CASE-FILE** starts at step 212 in Fig. 7B.

Intersections for each case of the Case-file with the current GPList is obtained in the remaining steps 204 to 211. A first entry from the Case-file is taken in step 204 and used as an input case when calling sub-routine **GEN-CASE-GPList** in step 205. Sub-routine **GEN-CASE-GPList** finds the generalization patterns between the input-case and items in the GPList and starts at step 220 in Fig. 7C. Sub-routine **GEN-CASE-EXAMINED-CASES** is called in step 206 for finding generalization patterns between the input-case and the cases in the Examined-case-list. Sub-routine **GEN-CASE-EXAMINED-CASES** starts at step 310 in Fig. 7H. In step 207, the input-case is added to the Examined-case-list. Cases (Cj) are therefore progressively moved from the Case-

file to the Examined-case-list until the Case-file is empty. Step 208 determines whether Case-file has any more entries. If there are any remaining entries in the Case-file, the procedure **MAIN** proceeds to step 209 where the next entry from Case-file is used as input-case and the procedure **MAIN** continues to step 205.

After step 208 determines that the Case-file is empty, the procedure **MAIN** produce generalization patterns in the GPList as an output in step 210 and ends in step 211.

Referring now to Fig. 7B wherein the sub-routine **UPDATE-CASE-FILE** is shown. This sub-routine includes the new cases produced as input in step 201 in procedure **MAIN** and into them into the current Case-file. It receives the entries from the list of new cases in step 213 and taking a next new case from the list of new cases in step 214, the next new case is added onto the end of the Case-file. Step 216 determines whether all the new cases has been dealt with and proceeds to steps 217 and 215 where the next entry is added to the Case-file. An updated Case-file is produced as output in step 219 and the sub-routine **UPDATE-CASE-FILE** ends in step 219.

Referring to Fig. 7C , the sub-routine **GEN-CASE-GPLIST** is to find all generalization patterns between the input-case and all items in the GPList. It further updates the GPList with all new generalization patterns between the input-case and the current GPList.

GEN-CASE-GPLIST starts in step 220 and receives the input case and the current GPList as inputs in step 221. It is determined in step 222 whether the current GPList is still empty. If the current GPList is empty, then no generalization patterns between the input-case and the GPList can exist and the sub-routine returns in step 236.

With items in the GPList, the sub-routine continues to step 223 wherein all generalization patterns between the input-case and all items in the GPList are found by calling sub-routine **G_List_GEN**. The generalization patterns are put in a **G_List**, which contains potential generalization patterns between the input-case and the GPList. The sub-routine **G_List_GEN** starts at step 240 in Fig. 7D.

It is determined in step 224 whether the **G_List** is empty. If the **G_List** is empty, then no generalization patterns between the input-case and the GPList was found by sub-routine **G_List_GEN** and the sub-routine **GEN-CASE-GPLIST** returns in step 236.

With items in the **G_List**, the sub-routine **GEN-CASE-GPLIST** continues to step 225 where sub-routine **UG_List_GEN** is called. Subroutine **UG_List_GEN** starts at step 260 in Fig. 7E, and forms a unique generalisation pattern list, **UG_List**, from the **G_List**.

A first item from the UG_List, UG_Item, is retrieved and the intersection, First_Intersection, from the UG_Item is extracted in step 226. A first item from the GPList, GPList_Item, is retrieved and the intersection, Second_Intersection, from the GPList_Item is extracted in step 227.

Step 228 calls sub-routine **INT_MATCH** to determine whether First_Intersection and Second_Intersection match. Sub-routine **INT_MATCH** starts at step 280 in Fig. 7F. If step 228 finds that First_Intersection and Second_Intersection match, the occurrence of GPList_Item is made equal to that of UG_Item in step 229. The sub-routine continues to step 233.

If step 228 finds that First_Intersection and Second_Intersection do not match, then step 230 determines whether all items of the GPList have been considered. If items remain, the next item in the GPList is retrieved with its intersection as Second_Intersection in step 231, before step 228 again determines whether First_Intersection and Second_Intersection match. If step 230 determined that all items in the GPList have been considered, then the UG_Item is added to the GPList in step 232 and the sub-routine continues to step 233.

If step 233 determines that all items in the UG_List have not been considered, then the next item in the UG_List is retrieved with its intersection as First_Intersection in step 234 and followed by step 227. Alternatively, if step 233 determines that all items in the UG_List have been considered, the sub-routine **GEN-CASE-GPLIST** outputs the GPList in step 235 and returns in step 236.

Referring to Fig. 7D, sub-routine **G_List_GEN** is described, which determines all generalisation patterns between the input-case with all items in the GPList. Starting at step 240, it receives as inputs the input-case and the GPList. The inputs are obtained in step 241. A first generalisation pattern, GPList_Item is obtained from the GPList in step 242, a first feature from the input case is retrieved in step 243 and a first feature from the intersection part of the GPList_Item is retrieved in step 244. Step 245 determines whether the retrieved feature from the input-case is the same as the retrieved feature from the GPList_Item. If no match is found in step 245, step 246 determines whether all features from the GPList_Item has been dealt with. With more features in the GPList_Item remaining, step 255 retrieves a next feature from the GPList_Item and continues to step 245. If all the features in the GPList_Item were considered, the sub-routine continues to step 247.

If step 245 responds in the affirmative, step 252 determines whether a new generalization pattern has been created and create one in step 253 if required, or if it was already done, proceeds to step 254, where the shared feature is added to the new generalization pattern. The sub-routine continues to step 247 where it is determined whether all features from the input-case have been dealt with. If there are remaining features in the input-case, step 256 retrieves the next feature from the input-case.

If step 247 determined that all the features from the input-case were considered, step 248 determines whether a new generalisation pattern was created in step 252. If the generalisation pattern already existed in the intersection parts of the GPList, the sub-routine continues to step 249 where the occurrence of the new generalisation pattern is given a value of the GPList Item that has the same intersection plus 1. In step 250 the new generalisation pattern is added to the G_List.

If step 248 determined that the new generalisation pattern does not already exist in the GPList, then the sub-routine continues to step 251.

Step 251 determines whether all items from the GPList have been considered. If an item remains in the GPList, the sub-routine continues to step 257 where the next item in the GPList is retrieved and step 243 is executed. Alternatively, with all items in the GPList considered, the sub-routine G_List_GEN returns in step 259 after producing the new G_List as output in step 258.

Referring to Fig. 7E, a sub-routine **UG_List_GEN** is shown that procedure forms a unique generalisation pattern list, **UG_List**. Starting in step 260, the sub-routine receives in step 261 the G_List as input. In step 262 the first generalisation pattern is copied from the G_List into the **UG_List**. A first item from the G_List, **G_List_Item**, is retrieved and the intersection, **First_Intersection**, from the **G_List_Item** is retrieved in step 263. A first item from the **UGList**, **UGList_Item**, is retrieved and the intersection, **Second_Intersection**, from the **UGList_Item** is retrieved in step 264.

Step 265 calls sub-routine **INT_MATCH** to determine whether **First_Intersection** and **Second_Intersection** match. Sub-routine **INT_MATCH** starts at step 280 in Fig. 7F. If **First_Intersection** and **Second_Intersection** match, the higher occurrence of the two items, **G_List_Item** and **UG_List_Item**, is determined and saved as the occurrence of **UG_List_Item** in steps 269 and 270. The sub-routine continues to step 268.

If **First_Intersection** and **Second_Intersection** do not match, step 266 determines whether all items of the **UG_List** have been considered. If items remain, the next item in the **UG_List** is retrieved with its intersection as **Second_Intersection** in step 271, before

step 265 again determines whether **First_Intersection** and **Second_Intersection** match. If step 266 determined that all items in the **UG_List** have been considered, then the **G_List_Item** is added to the **UG_List** in step 267 and the sub-routine continues to step 268.

If step 268 determines that all items in the **G_List** have not been considered, then the next item in the **G_List** is retrieved with it's intersection as **First_Intersection** in step 263 and followed by step 264. Alternatively, if step 268 determines that all items in the **G_List** have been considered, the sub-routine **UG_LIST_GEN** outputs the **UG_List** in step 273 and returns in step 274.

Referring to Fig. 7F, subroutine **INT_MATCH** is shown which check whether the **First_Intersection** and the **Second_Intersection**, obtained as inputs in step 281, are the same. Step 282 retrieves the first feature from **First_Intersection**, named **First-feature**, followed by step 283 where the first feature from **Second_Intersection**, named **Second-feature**, is retrieved. Step 284 calls sub-routine **FEATURE-SAME** to determine whether **First-feature** is the same as **Second-feature**. Sub-routine **FEATURE-SAME** starts at step 300 in Fig. 7G. If they are the same, then step 285 determines whether all the features of **First_Intersection** have been considered. If this is affirmative, then the sub-routine continues to step 286. If not, then step 296 retrieves the next feature from **First_Intersection**, names it **First-feature** and continues to step 283. However, if step 284 determined that **First-feature** is not the same as **Second-feature**, then step 290 determines whether all the features of the **Second_Intersection** have been considered. If this is affirmative, then the sub-routine **INT_MATCH** returns a "NO" in step 292. If features of the **Second_Intersection** remain, then step 291 retrieves the next feature from **Second_Intersection**, names it **Second-feature** and continues to step 284.

Step 286 retrieves the first feature from **Second_Intersection**, named **Second-feature**, followed by step 287 where the first feature from **First_Intersection**, named **First-feature**, is retrieved. Step 288 calls sub-routine **FEATURE-SAME** to determine whether **First-feature** is the same as **Second-feature**. If they are the same, then step 289 determines whether all the features of **Second_Intersection** have been considered. If this is affirmative, then the sub-routine **INT_MATCH** returns a "YES" in step 298. If not, then step 297 retrieves the next feature from **Second_Intersection**, names it **Second-feature** and continues to step 287. However, if step 288 determined that **First-feature** is not the same as **Second-feature**, then step 293 determines whether all the features of the **First_Intersection** have been considered. If this is affirmative, then the sub-routine

INT_MATCH returns a "NO" in step 295. If features of the **First_Intersection** remain, then step 294 retrieves the next feature from **First_Intersection**, names it **First-feature** and continues to step 288.

Referring now to Fig. 7G, sub-routine **FEATURE-SAME** is described, which checks whether two given features are the same. Therefore, in step 301 it receives as inputs, **First-feature** and **Second-feature**. Step 302 gets the attribute of **First-feature**, named **First-f-attribute**, and step 303 gets the attribute of **Second-feature**, named **Second-f-attribute**. Step 304 gets the value of **First-feature**, named **First-f-value**, and step 305 gets the value of **Second-feature**, named **Second-f-value**.

Step 306 determines whether **First-f-attribute** is the same as **Second-f-attribute**, and step 307 determines whether **First-f-value** is the same as **Second-f-value**. If the answers of both steps 306 and 307 are affirmative, then sub-routine **FEATURE-SAME** returns a "YES" in step 309. If any one of the answers of steps 306 and 307 is negative, then sub-routine **FEATURE-SAME** returns a "NO" in step 308.

Sub-routine **GEN-CASE-EXAMINED-CASES** starts in step 310 in Fig. 7H. This sub-routine finds generalisation patterns between the input-case from **Case-file** and cases in the **Examined-case-list**, which it receives as inputs in step 311. Step 312 determines whether the **Examined-case-list** is empty and returns in step 313 if this is affirmative. If the **Examined-case-list** has items, the sub-routine continues to step 314 where the first case from the **Examined-case-list** is retrieved. Step 315 calls subroutine **GET-GEN-PATTERN** to calculate a generalisation pattern, **Gen-pattern**, between the input-case and the case from the **Examined-case-list**. Subroutine **GET-GEN-PATTERN** starts at step 330 in Fig. 7I.

Step 316 determines whether a **Gen-pattern** has been found. If a **Gen-pattern** has been found, the sub-routine determines in step 317 whether the **Gen-pattern** match any item in the **GPList** by calling sub-routine **IF-MATCH**. Sub-routine **IF-MATCH** starts at step 350 in Fig. 7J. If the **Gen-pattern** does not match any item in the **GPList**, then step 318 adds **Gen-pattern** to **GPList** as a new item and continues to step 319. If step 317 find that the **Gen-pattern** match an item in the **GPList**, then the sub-routine continues to step 319 where it is determined whether all cases in the **Examined-case-list** have been considered.

If step 319 determines that cases in the **Examined-case-list** remain to be considered, then step 320 retrieves the next case from the **Examined-case-list** and

continues to step 315. Alternatively, step 321 outputs the GPList and the sub-routine **GEN-CASE-EXAMINED-CASES** returns in step 322.

Referring to Fig. 7I, wherein sub-routine **GET-GEN-PATTERN** for identifying a generalization pattern, Gen-pattern, between an input-case from the Case-file and one case from the Examined-case-list, is shown. The sub-routine starts in step 330 by obtaining input-case and a case from the Examined-case-list as inputs in step 331. It takes these two cases as input and compares their features. If some features are shared by the two cases, then these shared features will be included in the intersection part of Gen-pattern. The occurrence in Gen-pattern will be 2. Otherwise, if no features are shared between the cases, an empty Gen-pattern will be produced as output.

Step 332 retrieves the first feature from the input-case, named First-feature, followed by step 333 where the first feature from the case from the Examined-case-list, named Second-feature, is retrieved. Step 334 calls sub-routine **FEATURE-SAME** to determine whether First-feature is the same as Second-feature. Sub-routine **FEATURE-SAME** starts at step 300 in Fig. 7G. If they are the same, then step 335 would keep this feature in the intersection part of Gen-pattern and proceeds to step 338. If step 334 determined that the features were not the same, then step 336 determines whether all the features of the case from the Examined-case-list have been considered. If this is affirmative, then the sub-routine continues to step 338. If not, then step 337 retrieves the next feature of the case from the Examined-case-list, names it Second-feature and continues to step 334. Step 338 determines whether all features of the input-case have been considered. If this is affirmative, then the sub-routine continues to step 340. However, if step 338 determined that all the features of the input-case have not been considered, then step 339 retrieves the next feature of the input-case, names it First-feature and continues to step 333.

Step 340 determines whether the Gen-pattern is empty. If the Gen-pattern is not empty, then the occurrence of Gen-pattern is made 2 and the sub-routine **GET-GEN-PATTERN** returns in step 343 after producing as output Gen-pattern in step 342. If step 340 determines that Gen-pattern is empty, the sub-routine **GET-GEN-PATTERN** also returns in step 343.

Referring to Fig. 7J, sub-routine **IF-MATCH** is shown. This sub-routine is to check whether the intersection of the generalisation pattern of Gen-pattern matches the intersection of GPList. The sub-routine starts in step 350 and accepts Gen-pattern and GPList as inputs in step 351.

Step 352 extracts the intersection part of Gen-pattern and name it Gen-intersection. Step 353 retrieves the first item from GPList and the intersection part of this item is extracted and named GP-intersection in step 354.

Step 355 calls sub-routine **IF-SAME**, which starts at step 370 in Fig. 7K, to determine whether Gen-intersection and GP-intersection are the same. If a match is found, then the sub-routine **IF-MATCH** returns a "YES" in step 359. If a match was not found in step 355, then step 356 determines whether all the items of GPList have been considered. If this is affirmative, then the sub-routine **IF-MATCH** returns a "NO" in step 360. However, If step 356 determines that there are items of GPList that were not yet considered, then step 357 retrieves the next item from the GPList, step 358 extracts the intersection part of the item and name it GP-intersection and the sub-routine continues to step 355.

Referring to Fig. 7K, subroutine **IF-SAME** is shown which check whether Gen-intersection and GP-intersection, obtained as inputs in step 371, are the same. Step 372 retrieves the first feature from Gen-intersection, named First-feature, followed by step 373 where the first feature from GP-intersection, named Second-feature, is retrieved. Step 374 calls sub-routine **FEATURE-SAME** to determine whether First-feature is the same as Second-feature. Sub-routine **FEATURE-SAME** starts at step 300 in Fig. 7G. If they are the same, then step 378 determines whether all the features of Gen-intersection have been considered. If this is affirmative, then the sub-routine continues to step 379. If not, then step 388 retrieves the next feature from Gen-intersection, names it First-feature and continues to step 373. However, if step 374 determined that First-feature is not the same as Second-feature, then step 375 determines whether all the features of the GP-intersection have been considered. If this is affirmative, then the sub-routine **IF-SAME** returns a "NO" in step 292. If features of the GP-intersection remain, then step 376 retrieves the next feature from GP-intersection, names it Second-feature and continues to step 374.

Step 379 retrieves the first feature from GP-intersection, named First-feature, followed by step 380 where the first feature from Gen-intersection, named Second-feature, is retrieved. Step 381 calls sub-routine **FEATURE-SAME** to determine whether First-feature is the same as Second-feature. If they are the same, then step 385 determines whether all the features of GP-intersection have been considered. If this is affirmative, then the sub-routine **IF-SAME** returns a "YES" in step 386. If not, then step 387 retrieves the next feature from GP-intersection, names it First-feature and continues to

step 380. However, if step 381 determined that First-feature is not the same as Second-feature, then step 382 determines whether all the features of the Gen-intersection have been considered. If this is affirmative, then the sub-routine **IF-SAME** returns a “NO” in step 384. If features of the Gen-intersection remain, then step 383 retrieves the next feature from Gen-intersection, names it Second-feature and continues to step 381.

In Fig. 6, an example of a viewer profile 500 for a particular viewer is shown. The Viewer profile 500 may correspond to the GPList determined by method shown in Figs. 7A to 7K, performed by the learning module 39 of the avatar agent 37 associated with the particular viewer, with the GPList having the following intersections and occurrence values:

```

([Category = 'drama'], 6)
([Category = 'drama', Sub-category = 'social', Year_of_make = '1999'], 3)
([Category = 'drama', Sub-category = 'social', Start_Time = '2200', Year_of_make
    = '1999'], 2)
([Sub-category = 'social'], 8)
([Sub-category = 'social', Day = 'Thu'], 2)
([Category = 'movie'], 10)
([Sub-category = 'social', Day = 'Mon'], 2)
([Category = 'movie', Sub-category = 'social'], 5)
([Sub-category = 'social', Day = 'Wed'], 2)

```

The columns 501 in Fig. 6 represent the attributes of the intersections and the entries represent the values 502 of the attributes. Each row 503 in the viewer profile 500 represents an item in the viewer profile 500. Each item 503 in the viewer profile 500 also has a Date_Stamp entry. In the preferred implementation, the Date_Stamp value of the item 503 is the date of the most recent viewer program selection 68 that contributed to the intersections in the item 503. Alternatively, the Date_Stamp value may be the average of the dates of all viewer program selections 68 that contributed to the intersections in the item 503. For example, the item 503 in the third row has the following values and interpretation:

Attribute	Attribute-value	Interpretation
Category_ID	16	<i>Category</i> = 'drama'
Subcategory_ID	564	<i>Sub-category</i> = 'social'
Start_Time	2200	<i>Start_Time</i> = '2200'
Day	-1	Don't care
EPG_channel	-1	Don't care
Year_of_Make	1999	<i>Year_of_make</i> = '1999'

The occurrence of this intersection is 2 and the last time the viewer made a program selection 68 that contributed to this intersection was 24 April 1999. All entries with "-1", are interpreted as "Don't care".

Operations within the recommendation module 40 will now be described in more detail. The recommendation module 40, according to the preferred implementation, uses a number of tasks to compile the list of viewer recommendations 67. Various tasks are defined, with each task defining a unique combination of a manner of ordering the viewer profile 500, and particular Relevance Filters for filtering the ordered viewer profile.

The first aspect of each of the tasks, in accordance with the present implementation, is the manner in which the viewer profile 500 is ordered. The ordering arrangements may include:

- **WeightFactor Ordering** arranges the viewer profile entries 503 by *descending* occurrence value, ie. feature (fi) intersections occurring most frequently take precedence.
- **Specificity Ordering** arranges the viewer profile entries 503 by *descending* specificity level, ie. entries 503 which specify more program attribute-value pairs than others take precedence.
- **Generality Ordering** arranges the viewer profile entries 503 by *ascending* specificity level, ie. entries 503 which specify less program attribute-value pairs than others take precedence.
- **Recency Ordering** arranges the viewer profile entries 503 by *descending* Date_Stamp, ie. most recent entries 503 take precedence.

- **WeightRecency Ordering** arranges the viewer profile entries 503 by *descending* occurrence value, while preferring entries 503 having most recent Date_Stamp values when occurrence values are the same.
- **Specificity Weight Ordering** arranges the viewer profile entries 503 by *descending* specificity level, while preferring entries 503 having higher occurrence values when specificity levels are the same.
- **Generality Weight Ordering** arranges the viewer profile entries 503 by *ascending* specificity level, while preferring entries 503 having higher occurrence values when specificity levels are the same.
- **Recency Specificity Ordering** arranges the viewer profile entries 503 by *descending* Date_Stamp, while preferring more specific entries 503 when Date_Stamp values are the same.

The following possible orderings may be useful:

- **Specificity Recency Ordering** arranges the viewer profile entries 503 by *descending* specificity level, while preferring most recent entries 503 when specificity levels of entries are the same.
- **Generality Recency Ordering** arranges the viewer profile entries 503 by *ascending* specificity level, while preferring most recent entries 503 when specificity levels of entries are the same.

The second aspect of each of the tasks, in accordance with the present implementation, is the particular Relevance Filters applied to the ordered viewer profile, which includes:

- **TimeSlot Filter** verifies whether a viewer profile entry 503 is time- or day-specific and verifies the specific time and/or day with the day and time for which recommendations are requested.

Condition to reject Viewer profile entries 503: Start_Time is specified in the viewer profile entry 503, but is not equal to the time for which recommendations are requested or Day is specified in the viewer profile entry 503, but is not equal to the day for which recommendations are requested. The TimeSlot Filter is included in all filter operations.

- **WeightFactor Filter** verifies whether a viewer profile entry 503 has an occurrence value that is higher than a current average occurrence value of all the viewer profile entries 503.
Condition to reject Viewer profile entries 503: Occurrence value \leq average occurrence value.
- **Popularity Filter** verifies whether a viewer profile entry 503 corresponds to a (currently) popular event/program, eg. Football World Cup, Celine Dion concert.
Condition to reject viewer profile entries 503: When the POPULARITY value is false.
- **Urgency Filter** verifies whether a viewer profile entry 503 may be met by an urgent program, eg. special news/progress reports, weather announcements (tornado warnings, etc.), press conferences, nation addresses and so on.
Condition to reject viewer profile entries 503: When the URGENCY value is false.
- **LifeStyle Filter** verifies whether a viewer profile entry 503 corresponds to a (known) type of a lifestyle, eg., Midday movies, 7pm news, or Saturday night music shows.
Condition to reject Viewer profile entries 503: When the LIFESTYLE value is false.

By combining filter characteristics together, other filters can be obtained. For example,

- **PopularLifeStyle Filter** verifies whether a viewer profile entry 503 corresponds to a (known) type of a lifestyle and (currently) popular event/program, eg., Celine Dion concert on Saturday night.

Such combinations are significantly restrictive, and a task filtering ordered viewer profile entries with a combined filter may reject most if not all viewer profile entries 503. Therefore, it is expected that only a small proportion of viewers (most demanding ones) may value a task that includes such a filter.

Certain of the tasks provide recommendations due to more established Viewer profile entries 503 (gradual change), whereas other tasks attempt to produce viewer recommendations based on a sudden change in the viewer's viewing habits.

Using combinations of both the first (ordering) and second (filtering) aspects of the tasks, a number of useful tasks may be defined including:

Gradual Change Tasks

Basic Task = Weight Factor Ordering + TimeSlot Filter
(broad-minded viewer)

DieHard Task = Specificity Weight Ordering + WeightFactor Filter
(old habits die hard)

OldFavourites Task = Generality Weight Ordering + WeightFactor Filter
(original recipe)

BestPick Task = Specificity Weight Ordering + Popularity Filter
(choosy viewer)

Prime Time Task = Weight Recency Ordering + LifeStyle Filter
(my TV – my castle)

Sudden Change Tasks

HotTopic Task = Recency Specificity Ordering + Popularity Filter
(sudden change due to popularity change)

NewLife Task = Recency Specificity Ordering + LifeStyle Filter
(sudden change due to life style change)

WatchDog Task = Recency Specificity Ordering + Urgency Filter
(sudden change due to crisis situation).

Because different viewers may be satisfied with viewer recommendations produced as a result of the execution of different tasks, each task has an associated viewer specific Task_Value, indicating the particular viewer's satisfaction with the viewer recommendations 67 provided by the particular task. The Task_Value is also stored in the viewer profile database 23 and each task, represented by a unique Task_ID, may be stored as follows:

Task_ID	Number
Task_Value	Number
Task_Type	gradual or sudden change

Generally, an optimal set of viewer recommendations 67 is obtained as a result of a balance between various tasks. In the preferred implementation, a combination of gradual change tasks and sudden change tasks is used to compile the viewer recommendations 67.

Referring to Fig. 8, when the recommendation module 40 is instructed by the avatar manager 38 to provide television program recommendations, the procedure

RECOMMEND is started in step 801. As inputs the procedure **RECOMMEND** receives:

- The Viewer profile 500 from the viewer profile database 23, which includes viewer profile entries 503. Each viewer profile entry 503 includes the intersecting features and the occurrence count;
- A list of program entries and their features (fi) from which to select viewer recommendations 67. This list was retrieved by the avatar manager 38 based on the time/date combination provided by the viewer for which viewer recommendations are required; and
- a predetermined maximum number of recommended programs;

Initially the list of viewer recommendations is empty. If it is determined in step 802 that all the viewer profile entries 503 have been examined or that there are already the predetermined maximum number of viewer recommendations or that the list of program entries is empty, then the procedure continues to step 805 where the list of viewer recommendations are passed to the avatar manager 38. Otherwise, the procedure would continue to step 803, wherein a next viewer profile entry 503 is selected according to a selection criterion. The selection criterion may be one of the following:

- Choose the next viewer profile entry 503 with the lowest occurrence. (i.e. start from a more specific match).
- Choose the next viewer profile entry 503 with the highest occurrence (i.e. start from the most general match)
- Choose the next viewer profile entry 503 with the greatest number of features in the intersection (i.e. the most specific match)

In step 804, all program entries 60 in the list of program entries whose features (fi) includes those in the viewer profile entry 503 chosen in step 803, are moved from the list of program entries to the list of viewer recommendations. The procedure goes back to step 802.

Referring to Fig. 9A where an alternative procedure for providing television program recommendations is shown. Procedure **RECOMMEND** is started in step 601. In step 602 the procedure **RECOMMEND** receives as inputs:

- The Viewer profile 500 from the viewer profile database 23, which includes viewer profile entries 503. Each viewer profile entry 503 includes the intersecting features, the Date_Stamp and the occurrence count;

- The viewer specific Task_Values;
- A list of program entries and their features (fi), retrieved from the EPG databases 22; and
- A predetermined desired number of recommended programs R.

In step 603 a counter l is set to a value of 1. A variable K is also set, where K represents the number of tasks used by the recommendation module to provide viewer recommendations 67. In the preferred implementation, the gradual change task having a maximum Task_Value and all the sudden change tasks are used. Hence, the variable K is set as the number of sudden change tasks plus 1.

This is followed by step 604 where a task, identified by its Task_ID, is chosen. With the counter l equal to 1, step 604 chooses the gradual change task with a maximum Task_Value. If the counter l is larger than 1, the next sudden change task is chosen.

Step 605 sets a variable α_l . The variable α_l represents the desired portion of viewer recommendations that the recommendation module 40 would determine that would be made up from the task chosen in step 604. Accordingly, the variable α_l is set as $1/(2n)$, with n being 1 or K-1 for gradual and sudden change tasks respectively.

Step 606 calls sub-routine **REASON** for applying the chosen task and for producing viewer recommendations 67. Sub-routine **REASON** starts at step 650 in Fig. 9B.

Step 607 determine whether the **REASON** sub-routine has produced at least 1 recommendation for the chosen task. If the **REASON** sub-routine was unable to produce the at least 1 recommendation, then the intersecting features of the viewer profile entry 503 are relaxed in step 609. In the preferred implementation, all intersections are ignored except for category and sub-category, if they exist as intersections. The procedure **RECOMMEND** returns to step 606 where the sub-routine **REASON** is called again with the relaxed features.

If step 608 determines that the features of the viewer profile entry 503 were already relaxed, then the procedure **RECOMMEND** continues to step 621.

If step 607 determines that the **REASON** sub-routine was able to produce at least one recommendation for the chosen task, then step 621 determines whether the counter l is still smaller than variable K. If the counter l has a value of K, then recommendations for the desired tasks have been produced and the procedure **RECOMMEND** continues to step 623 where the SetRecommendations message, containing a list of recommended

programs, is passed to the avatar manager 38, from where it is sent to the DTV agent 36. Alternatively, the recommendation module 40 may send the **SetRecommendations** message directly to the DTV agent 36.

However, if step 621 determines that the counter l is smaller than variable K , then viewer recommendations for certain tasks must still be produced. Therefore, step 622 increments the counter l and the procedure **RECOMMEND** returns to step 604 for determining viewer recommendations for a next task.

Referring to Fig. 9B wherein the sub-routine **REASON** for applying the chosen task and for producing viewer recommendations is shown. Sub-routine **REASON** starts in step 650.

Step 652 orders the viewer profile entries 503 by applying the ordering aspect of the task selected in step 604 of the **RECOMMEND** procedure. For example, if the gradual change task DieHard task was selected, then the viewer profile entries 503 are ordered by the Specificity Weight Ordering which arranges the viewer profile entries 503 by descending specificity level, while preferring entries 503 having higher occurrence values when specificity levels are the same. Step 653 identifies the filter, specified by chosen task. For example, with the DieHard task, step 653 identifies the filter WeightFactor Filter, which verifies whether a viewer profile entry 503 has an occurrence value that is higher than a current average occurrence value of all the viewer profile entries 503.

Step 655 sets a viewer profile entry counter j to a value of 1 and a recommendations counter r to a value of 0. In step 657 it is determined whether the j -th ordered viewer profile entry 503 satisfies the chosen filter from step 653. It is again noted that the TimeSlot Filter is always included in all filter operations whereby all viewer profile entries 503 with specific Start_Time, Start_Day or Start_Date (ie not equal to -1: don't care), and where the specific Start_Time Start_Day or Start_Date is not equal to the current time, current day or current date respectively, are removed.

If step 657 determines that the j -th ordered viewer profile entry 503 does not satisfy the chosen filter, then the sub-routine **REASON** continues to step 677 for determining whether the viewer profile counter j is still smaller than the total number of viewer profile entries 503, N .

Alternatively, if step 657 determines that the j -th ordered viewer profile entry 503 does satisfy the chosen filter, then the sub-routine **REASON** continues to step 660 where

a dynamic rule is build as EPG database filter by calling sub-routine **BUILD_DYNAMIC_RULE**. The sub-routine **BUILD_DYNAMIC_RULE** is shown in Fig. 9C and starts in step 700.

Step 662 applies the rule that was built in the sub-routine **BUILD_DYNAMIC_RULE** on the entries 60 of the EPG databases 22 to identify in EPG database entries 60 that satisfies the rule. This is followed by step 665 which calls the sub-routine **INTERPRET**. Sub-routine **INTERPRET** builds a map of recommendations and is shown in Fig. 9D.

Sub-routine **PRODUCE** is called in step 667 which produces an ordered list of viewer recommendations from the map of recommendations. Sub-routine **PRODUCE** is shown in Fig. 9E and starts in step 750.

Step 672 calculates the recommendations counter r to be the previous count plus the number m of recommendations produced by sub-routine **PRODUCE** called in step 667. Step 675 determines whether the recommendations counter r is smaller than the portion α_i of the predetermined desired number **R** of recommended programs. If the determination of step 675 is negative, then a sufficient number of recommendations were created for that task and the sub-routine **REASON** continues to step 682. Alternatively, if step 675 determines that not enough recommendations were created, the sub-routine **REASON** continues to step 680 where the list of ordered recommendations, is joined in a ranked set **X**. This is done by, for each entry from the ordered list of recommendations, determining whether the entry already exists in the ranked set **X**. If the entry already exists, the entry is ignored and the next entry is considered. However, if the entry does not yet exist, then it is entered at the end of the ranked set **X**.

Step 677 then determines whether the viewer profile counter j is still smaller than the total number of viewer profile entries 503 and if so, directs the sub-routine **REASON** to step 658 where the counter j is incremented before directing the sub-routine back to step 657. This allows more recommendations to be determined by finding a next viewer profile entry 503 that meet the filter criteria.

From step 675, or from step 677 when all the viewer profile entries 503 have been considered, the sub-routine **REASON** returns to the procedure **RECOMMEND** in step 682.

Referring to Fig. 9C wherein the sub-routine **BUILD_DYMANIC_RULE** is shown. Sub-routine **BUILD_DYMANIC_RULE** builds a dynamic rule for the ordered viewer profile entry 503 that satisfied the chosen filter, and starts in step 700.

Step 702 sets a string *Y* as the source of the EPG program database 22. The string *Y*, and hence the EPG program database 22 is formulated in a string "tmp" in step 705. Steps 707 to 712 set variables day, hours and minutes as the date and time for which recommendations should be provided. This date and time were obtained from the DTV agent in the SendRecommendationRequest message. The variables day, hours and minutes are added to the string "tmp", allowing a 2 hour window around the hours variable.

Step 715 determines whether the intersecting features of the viewer profile entry 503 were relaxed by performing step 609. The sub-routine **BUILD_DYMANIC_RULE** continues to step 718 if the intersecting features of the viewer profile entry 503 were relaxed, where the category of the ordered viewer profile entry 503, if it is not equal to -1 (don't care), is added to the string "tmp". Similarly, in step 719 the sub-category of the ordered viewer profile entry 503 is added to the string "tmp" if it is determined not to be equal to -1 (don't care). The sub-routine then proceeds to step 720.

If step 715 determines that the intersecting features have not been relaxed, then step 717 determines whether all the attributes of the ordered viewer profile entry 503 have been processed. If so, then a string "rule" is set to be equal to the string "tmp" in step 720 before sub-routine **BUILD_DYMANIC_RULE** returns to the sub-routine **REASON** in step 729.

However, if there are unprocessed attributes of the ordered viewer profile entry 503, then steps 722 and 725 gets the next attribute whose value is not -1 (don't care) and add that attribute to the string "tmp". The sub-routine continues to step 717.

Referring now to Fig. 9D wherein the sub-routine **INTERPRET** is shown. Sub-routine **INTERPRET** builds a map of recommendations from the returned EPG database entries 60, and starts in step 730.

The sub-routine **INTERPRET** is initialised in step 732 by setting the map of viewer recommendations as empty, and setting a counter *c* as 1. Step 735 gets the *c*-th entry of the returned EPG database entries 60 and step 737 adds this entry to the map of recommendations according to the category of the entry. The map of recommendations will have as entries rows having the categories, followed by all the returned EPG database

entries 60 having a particular category as a feature. Each row would typically have a different number of fields.

Step 740 determines whether the variable *c* is smaller than the total of returned EPG database entries 60, or in other words, whether all the returned EPG database entries 60 have been entered into the map of recommendations. If it is determined in step 740 that all the returned EPG database entries 60 have not been added, then the variable *c* is incremented in step 742 and the sub-routine **INTERPRET** continues again to step 735.

However, if step 740 determines that all the returned EPG database entries 60 have been added to the map of recommendations, then step 745 returns the sub-routine **INTERPRET** to the subroutine **REASON**.

Referring to Fig. 9E wherein the sub-routine **PRODUCE** for producing in viewer recommendations is shown. Sub-routine **PRODUCE** starts in step 750.

The sub-routine **PRODUCE** is initialised in step 752 by setting the vectors of recommendations as empty, and setting a counter *h* as 1. Step 755 gets the *h*-th row from the recommendations map and step 757 retrieves a first component of the *h*-th row of the recommendations map as the category ID and a second component as all the returned EPG database entries 60 (or programs) with that category as a feature. Step 758 adds the first and second components together into a vector of categorised recommendations.

Step 760 determines whether the variable *h* is smaller than the total number of rows in the recommendations map, or in other words, whether all the rows in the recommendations map have been converted into vectors. If it is determined in step 760 that the recommendations map has not been fully converted, then the variable *h* is incremented in step 761 and the sub-routine **PRODUCE** continues again to step 755.

However, if step 760 determines that all the rows of the recommendations map have been converted, then step 762 returns the sub-routine **PRODUCE** to the subroutine **REASON**.

When the avatar manager 38 receives the message **SetProgramSelection** from the DTV agent 36 (step 13 of Table 2), indicating that the viewer has selected a particular program 68, the avatar manager 38 calls routine **ON_PROGRAM_SELECT**, which is shown in Fig. 9F and starts in step 770. Step 775 determines whether the selected program was within the first half of the ranked list of recommendations X. If this determination is found to be true, then a variable *inc*, which is a Task_Value increment, is

set to a positive value in step 776. Alternatively, the variable *inc* is set to a negative value in step 777.

After setting the variable *inc* in step 776 or step 777, step 782 adds the variable *inc* to the Task_Value of the Gradual Change task selected in step 604 of the **RECOMMEND** procedure. Therefore, if the selected program is in the upper half of the set of ranked recommendations X, then the viewer recommendation 67 are deemed to have been successful, and the Task_Value of the gradual task used by the recommendation module 40 is increased, thereby ensuring that it will again be used for future recommendations. However, if the selected program was listed in the lower half of the set of ranked recommendations X, the recommendation possibly was not optimal, and the Task_Value of the gradual task used by the recommendation module 40 is decreased.

In an alternative implementation, sub-routine **INTERPRET** records the Task_ID of the task(s) producing each of the recommendations. This allows the routine **ON_PROGRAM_SELECT** to only adjust the Task_Value of the task that produced the selected recommendation.

The methods of Figs 7A to 7K, and Fig. 8 or Figs. 9A to 9F, may be practiced using a conventional general-purpose computer system 100, such as that shown in Fig. 10 wherein the processes of Figs 7A to 7K, Fig. 8 and Fig. 9 may be implemented as software, such as an application program executing within the computer system 100. In particular, the steps may be instructions in the software that are carried out by the computer. The software may be stored in a computer readable medium, including the storage devices described below. The software is loaded into the computer from the computer readable medium, and then executed by the computer. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer preferably effects an advantageous apparatus in accordance with the implementations of the invention.

The computer system 100 comprises a computer module 102, input devices such as a keyboard 110 and mouse 112, output devices including a printer 108 and a display device 104.

The computer module 102 typically includes at least one processor unit 114, a memory unit 118, for example formed from semiconductor random access memory (RAM) and read only memory (ROM), input/output (I/O) interfaces including a video interface 122, and an I/O interface 116 for the keyboard 110 and mouse 112. A storage

device 124 is provided and typically includes a hard disk drive 126 and a floppy disk drive 128. A magnetic tape drive (not illustrated) may also be used. A CD-ROM drive 120 is typically provided as a non-volatile source of data. The components 114 to 128 of the computer module 102, typically communicate via an interconnected bus 130 and in a manner which results in a conventional mode of operation of the computer system 100 known to those in the relevant art. Examples of computers on which the implementations can be practised include IBM-PC's and compatibles, Sun Sparcstations or alike computer systems evolved therefrom.

Typically, the application program of the preferred implementation is resident on the hard disk drive 126 and read and controlled in its execution by the processor 114. Intermediate storage of the program may be accomplished using the semiconductor memory 118, possibly in concert with the hard disk drive 126. In some instances, the application program may be supplied to the viewer encoded on a CD-ROM or floppy disk and read via the corresponding drive 120 or 128, or alternatively may be read by the viewer from a network via a modem device (not illustrated). Still further, the software can also be loaded into the computer system 100 from other computer readable medium including magnetic tape, a ROM or integrated circuit, a magneto-optical disk, a radio or infra-red transmission channel between the computer module 102 and another device, a computer readable card such as a PCMCIA card, and the Internet and Intranets including email transmissions and information recorded on websites and the like. The foregoing is merely exemplary of relevant computer readable mediums. Other computer readable mediums may be practiced without departing from the scope and spirit of the invention.

The methods of Figs 7A to 7K, Fig. 8 and Figs. 9A to 9F may alternatively be implemented in dedicated hardware such as one or more integrated circuits performing the functions or sub functions of Figs 7A to 7K, Fig. 8 and Fig. 9A to 9F. Such dedicated hardware may include graphic processors, digital signal processors, or one or more microprocessors and associated memories.

The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiment being illustrative and not restrictive.

4. Brief Description of the Drawings

Fig. 1A is a schematic representation of a system;

Fig. 1B is a detailed representation of an avatar agent of the system in Fig. 1A;

Fig. 2 is an extract from a typical Electronic Program Guide;

Fig. 3 is an illustration of an animated character displayed on a display screen of the system in Fig. 1A;

Fig. 4 is an example of how recommendations may be presented to a viewer by the system in Fig. 1A;

Fig. 5 is an example of a selection made from the recommendations in Fig. 4;

Fig. 6 is an example of a viewer profile database;

Figs 7A to 7K are flow diagrams of a method performed by a learning module of the avatar agent illustrated in Fig. 1A, to learn common features from selections made by the viewer;

Fig. 8 is a flow diagram of a method performed by a recommendation module of the avatar agent illustrated in Fig. 1B, reasoning to best use the results from the learning module to make recommendations of programs to watch to the viewer;

Figs. 9A to 9F are flow diagrams of an alternative method of reasoning to best use the results from the learning module to make recommendations of programs to watch to the viewer; and

Fig. 10 is a schematic block diagram of a general-purpose computer upon which arrangements described can be practiced.

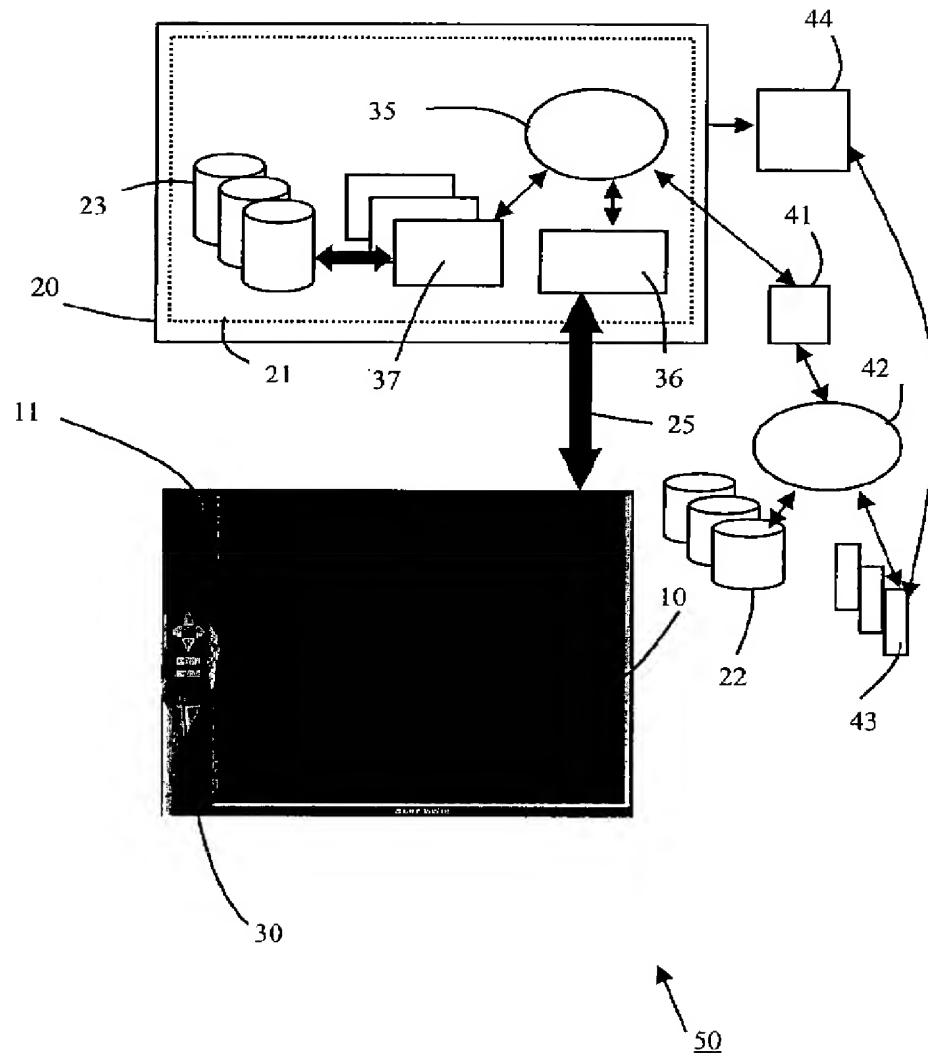


Fig. 1A

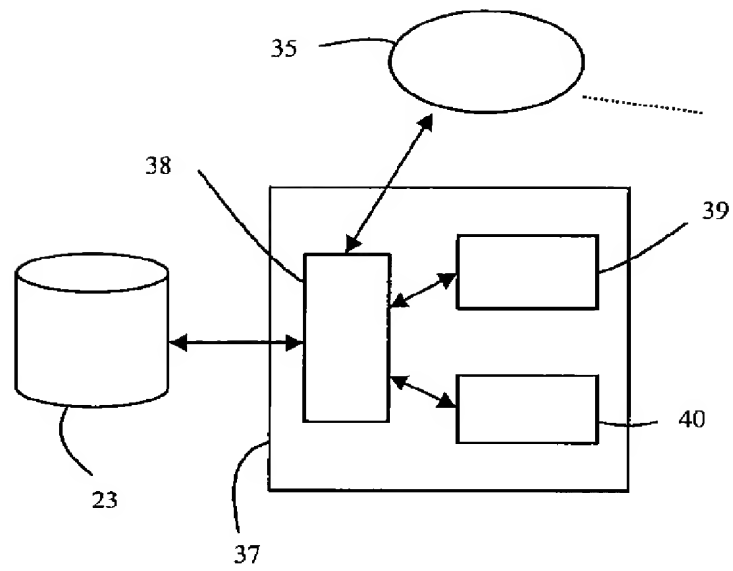


Fig. 1B

61

EPG ID	Program ID	Category ID	Sub-category ID	Title	EPG Channel	Day	Date	Start-Time	Exact Duration	Year Of Make
1	1	16	564	ferishitei (felicity) no seishun	Wow	Fri	4/9/99	20:00	50	1999
2	2	16	564	Suzuran	Nhk	Mon	4/5/99	8:15	15	1999
3	3	16	217	genroku ryoran	Nhk	Sun	4/4/99	20:00	45	1999
4	4	16	564	nichiyo gekijo/goddo nyu-su	Tbs	Sun	4/11/99	21:00	54	1999
5	5	16	563	abunai ho-kago	Asahi	Mon	4/12/99	20:00	54	1999
6	6	16	564	nippu suteikku (lip stick)	Fuji	Mon	4/12/99	21:00	54	1999
7	7	16	519	Romansu	Nihon	Mon	4/12/99	22:00	84	1999
8	8	16	188	furuhata ninzaburo	Fuji	Tue	4/13/99	21:00	69	1999
9	9	16	001	kizudarake no onna	Fuji	Tue	4/13/99	22:15	54	1999
10	10	16	564	happi ai to kando no monogatari	Tokyo	Wed	4/14/99	20:00	54	1999
11	11	16	563	Naomi	Fuji	Wed	4/14/99	21:00	54	1999
12	12	16	564	aka ga hashiru - to-gei seishun ki	Nhk	Wed	4/7/99	22:00	45	1999
13	13	16	519	semi daburu	Fuji	Wed	4/14/99	22:00	54	1999
14	14	16	188	Maiko san wa mitantei!	Asahi	Wed	4/15/99	20:00	54	1999
15	15	16	519	koi no kiseki	Asahi	Thu	4/15/99	21:00	69	1999

60

62

22

Fig. 2

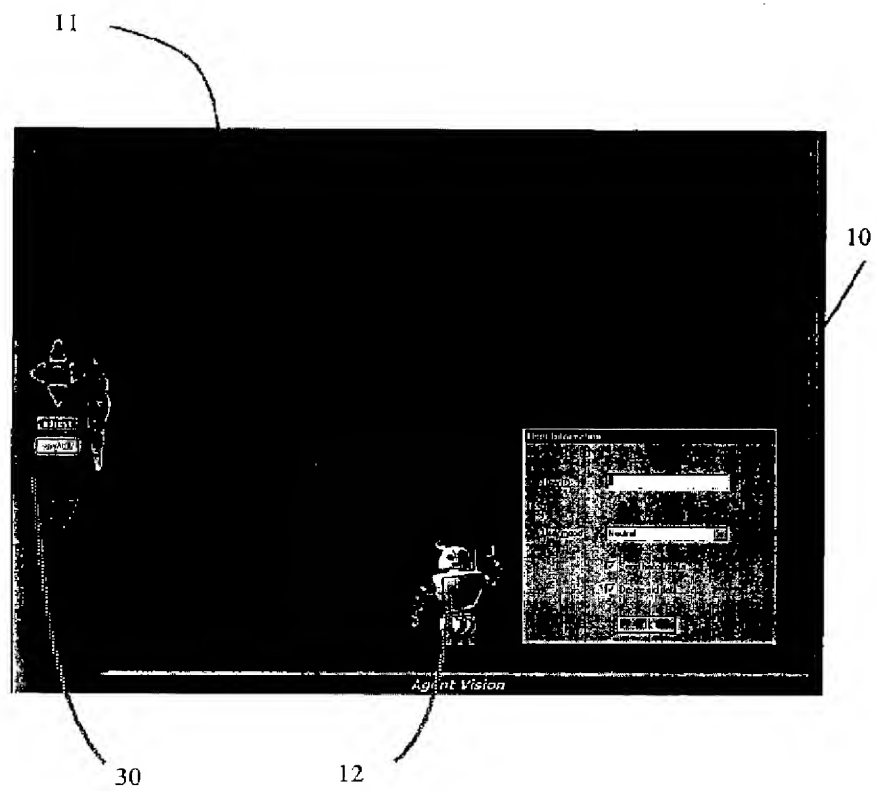


Fig. 3

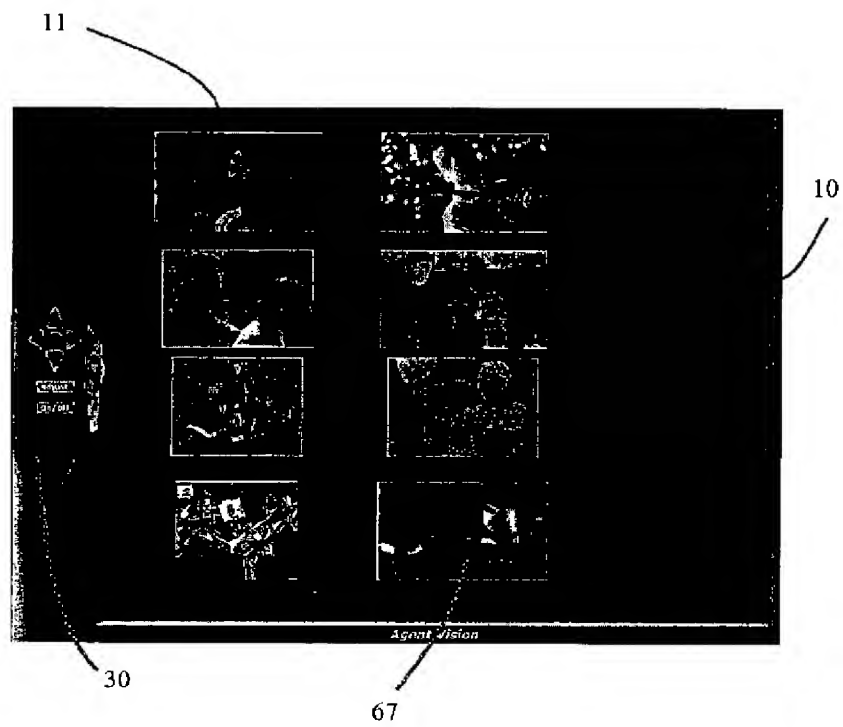


Fig. 4

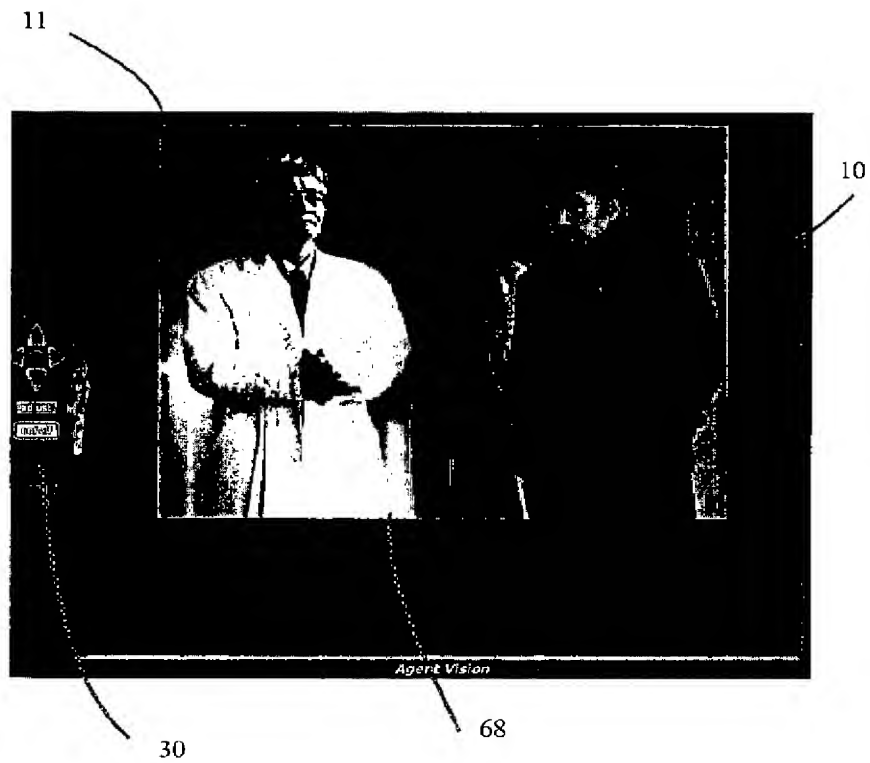


Fig. 5

500

502

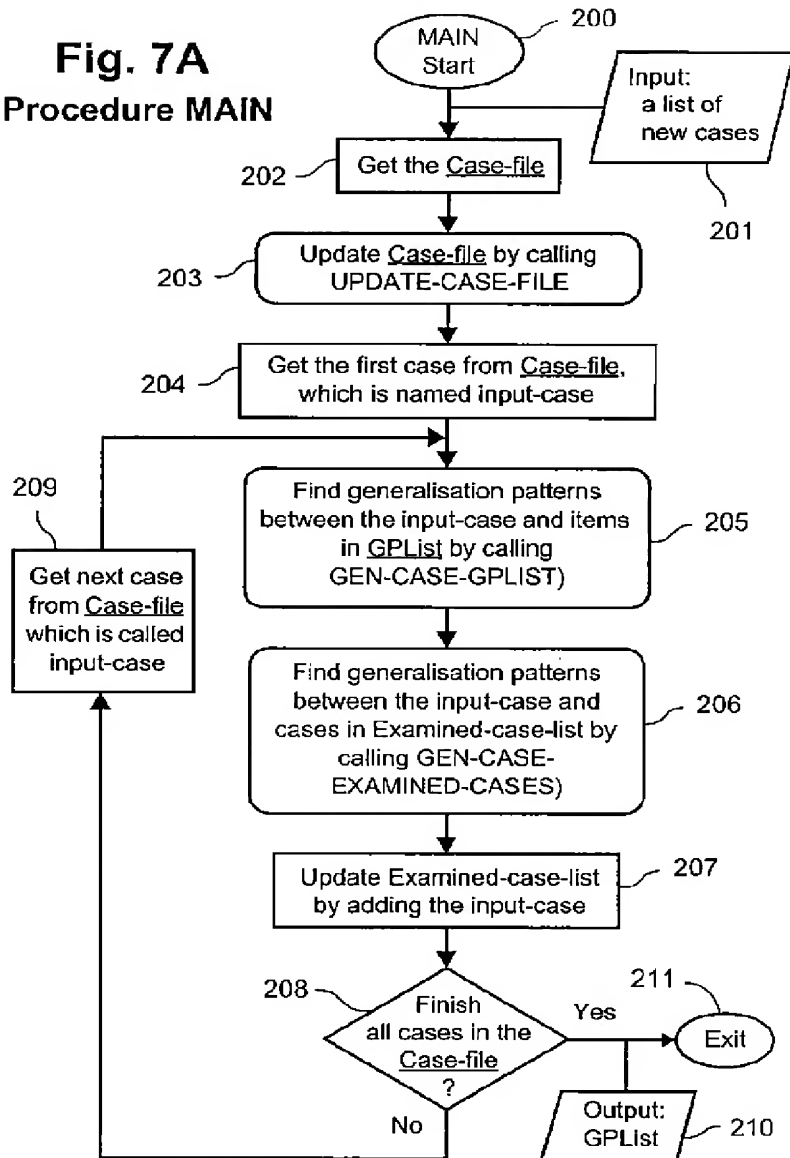
503

501

Category_ID	Subcategory_ID	Start_Time	Day	EPG_channel	Year_of_make	Occurrence
16	-1	-1	-1	-1	-1	6
16	564	-1	-1	-1	1999	3
16	564	2200	-1	-1	1999	2
-1	564	-1	-1	-1	-1	8
-1	564	-1	5	-1	-1	2
50	-1	-1	-1	-1	-1	10
-1	564	-1	2	-1	-1	2
50	564	-1	-1	-1	-1	5
-1	564	-1	4	-1	-1	2

Fig. 6

Fig. 7A
Procedure MAIN



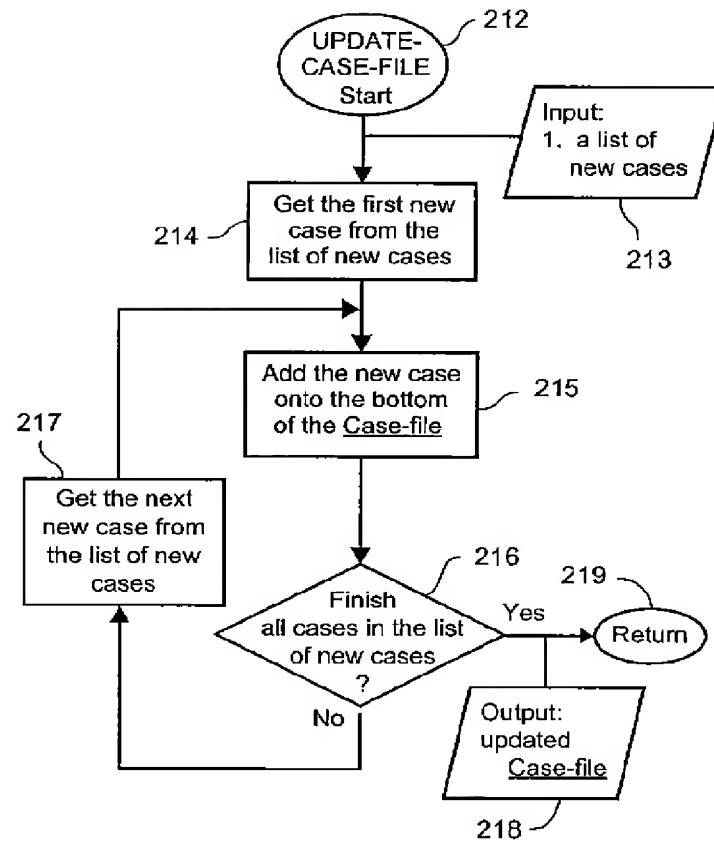


Fig. 7B
Sub-routine UPDATE-CASE-FILE

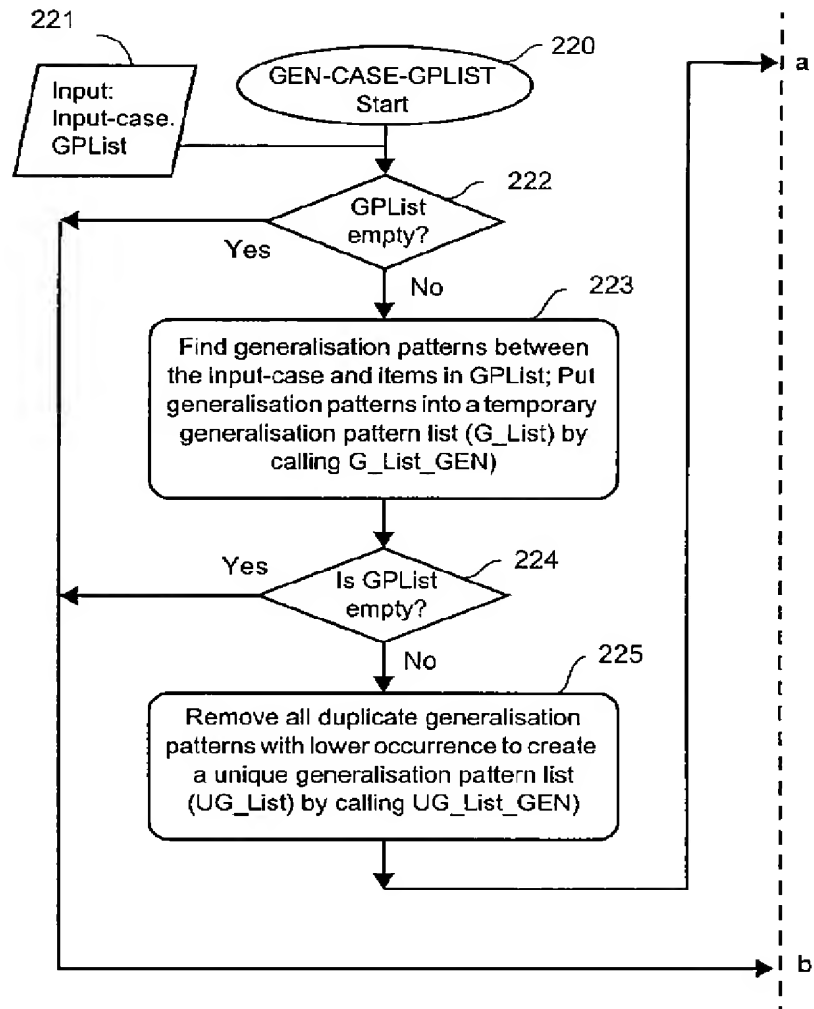


Fig. 7C
(a)

Fig. 7C
(b)

Fig. 7C (a)
Sub-routine
GEN-CASE-GPLIST

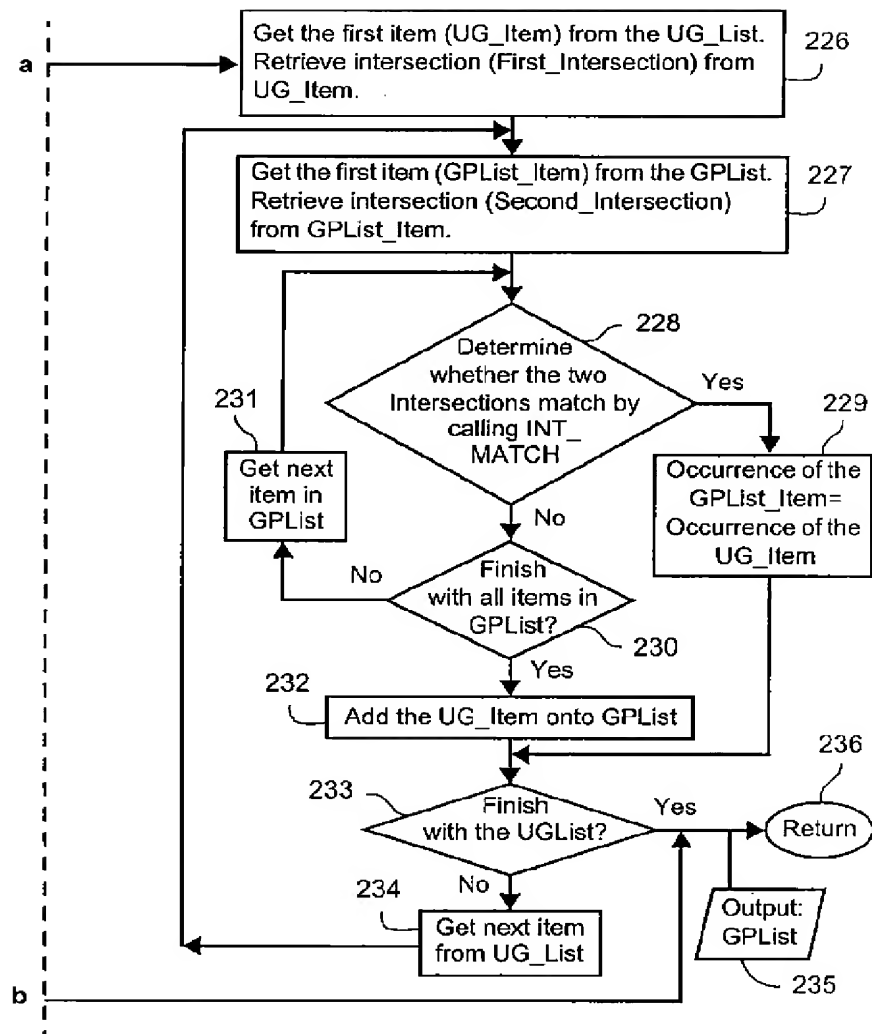
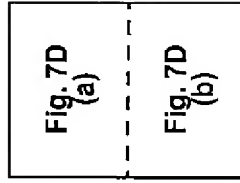
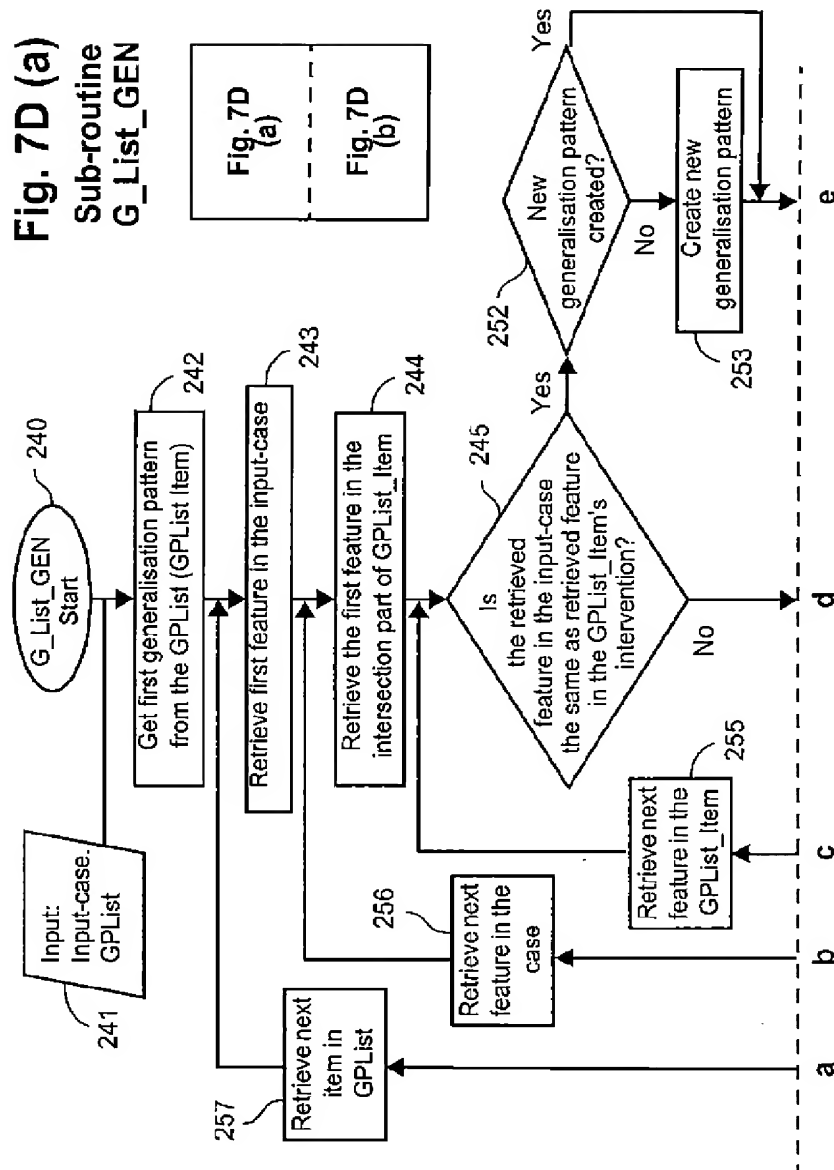
Fig. 7C
(a)Fig. 7C
(b)

Fig. 7C (b)
Sub-routine
GEN-CASE-GPLIST

Fig. 7D (a)
Sub-routine
G_List_GEN



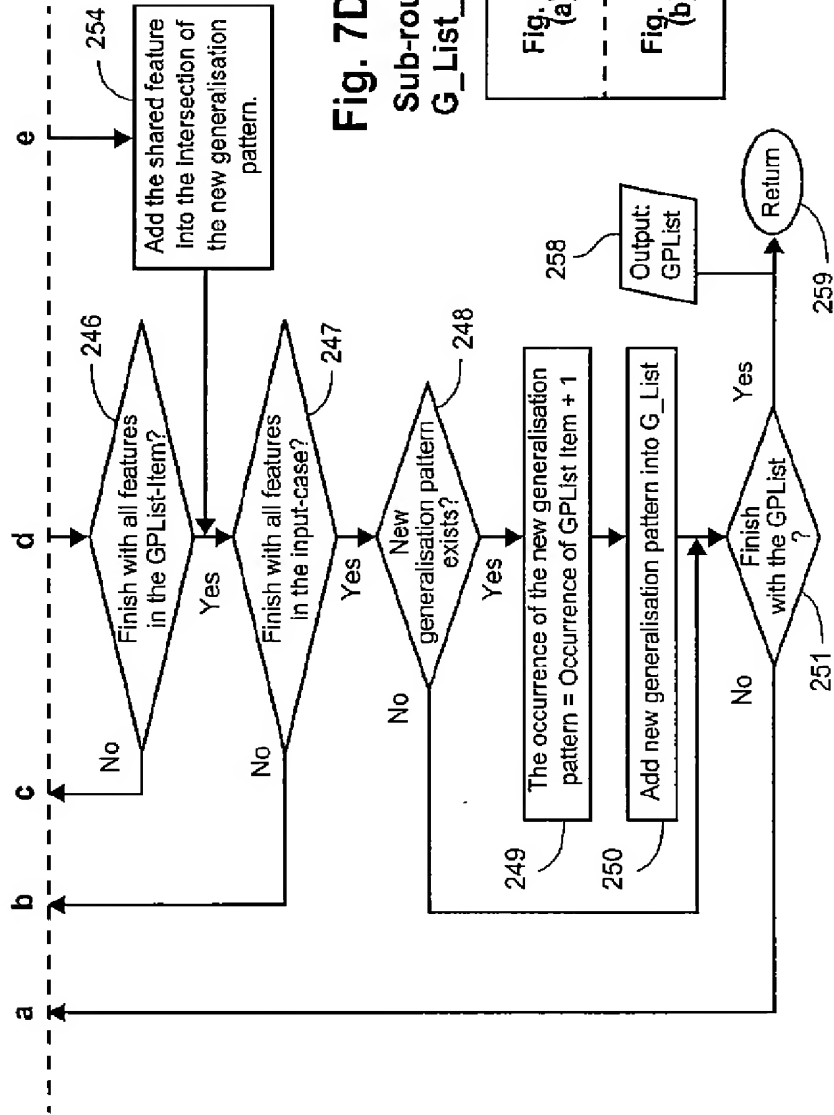


Fig. 7D (b)
Sub-routine
G_List_GEN

Fig. 7D
 (a)

Fig. 7D
 (b)

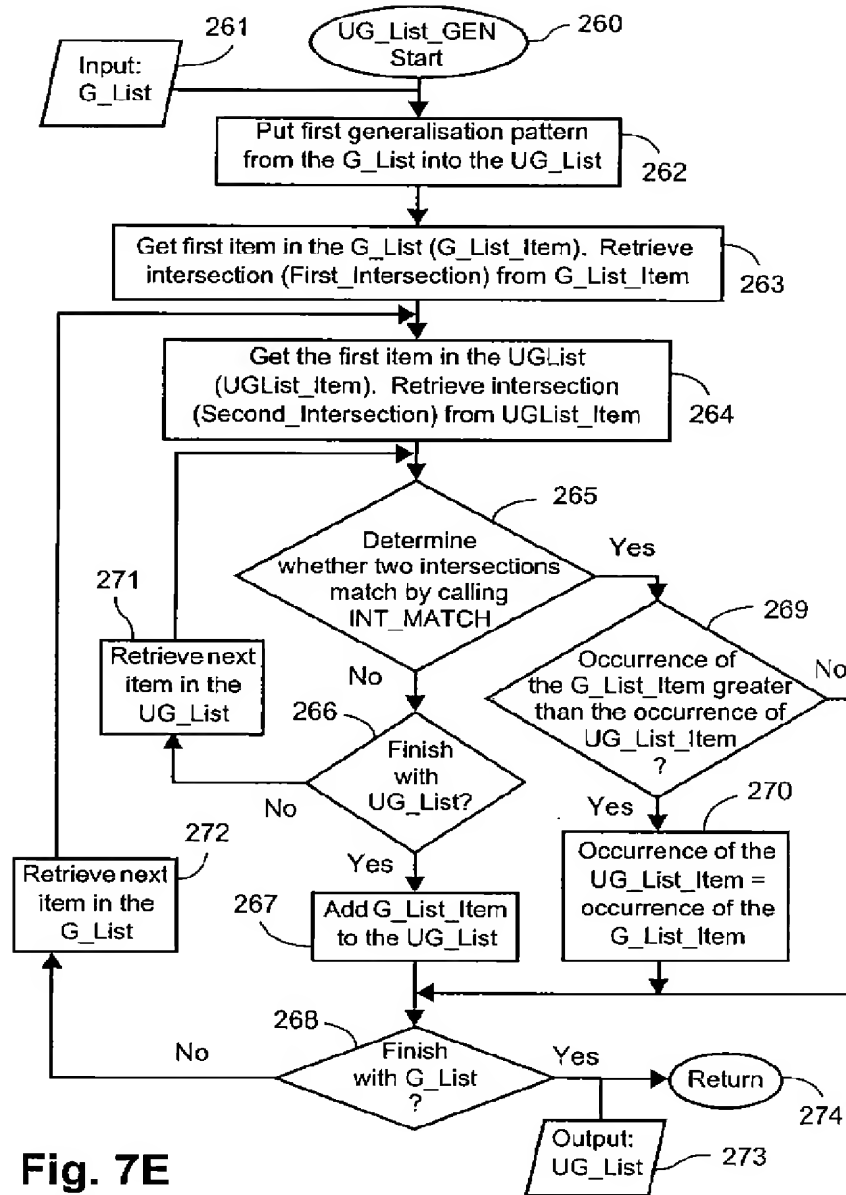


Fig. 7E
Sub-routine UG_List_GEN

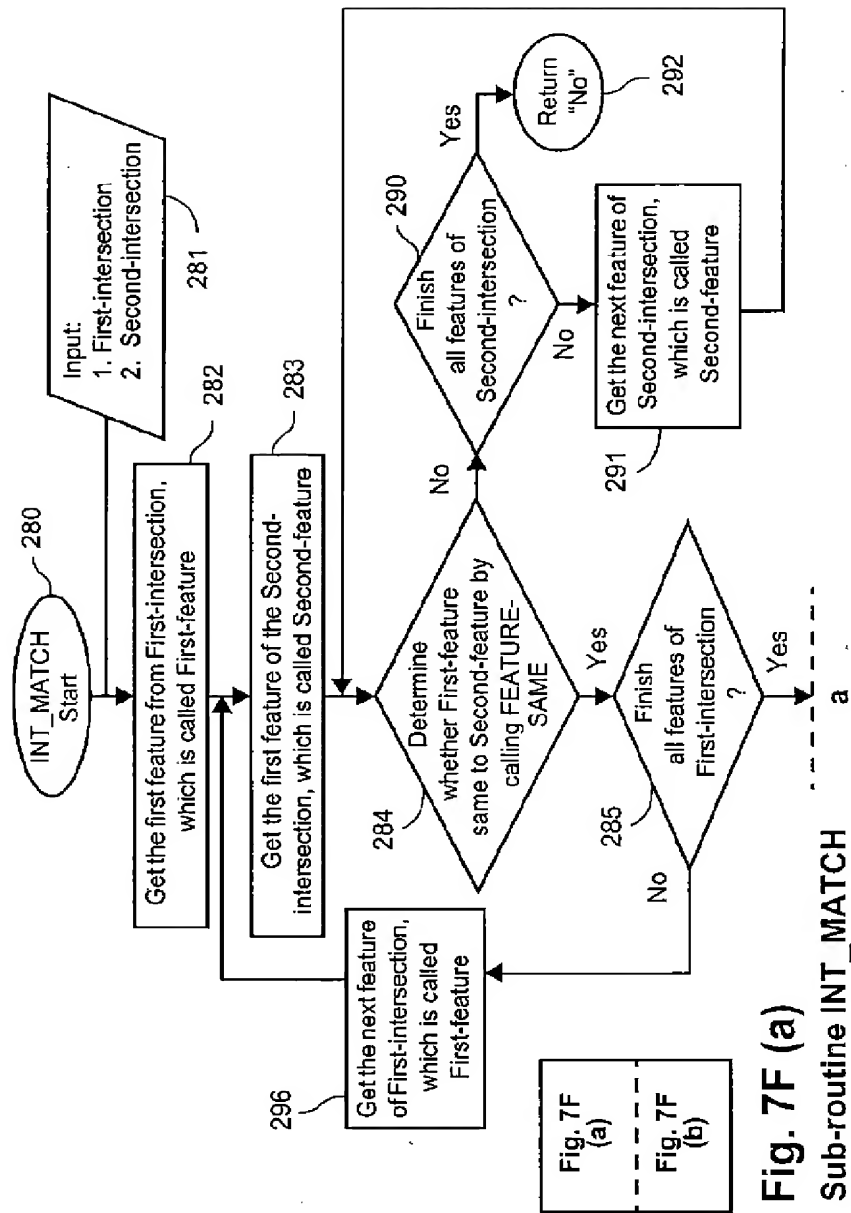
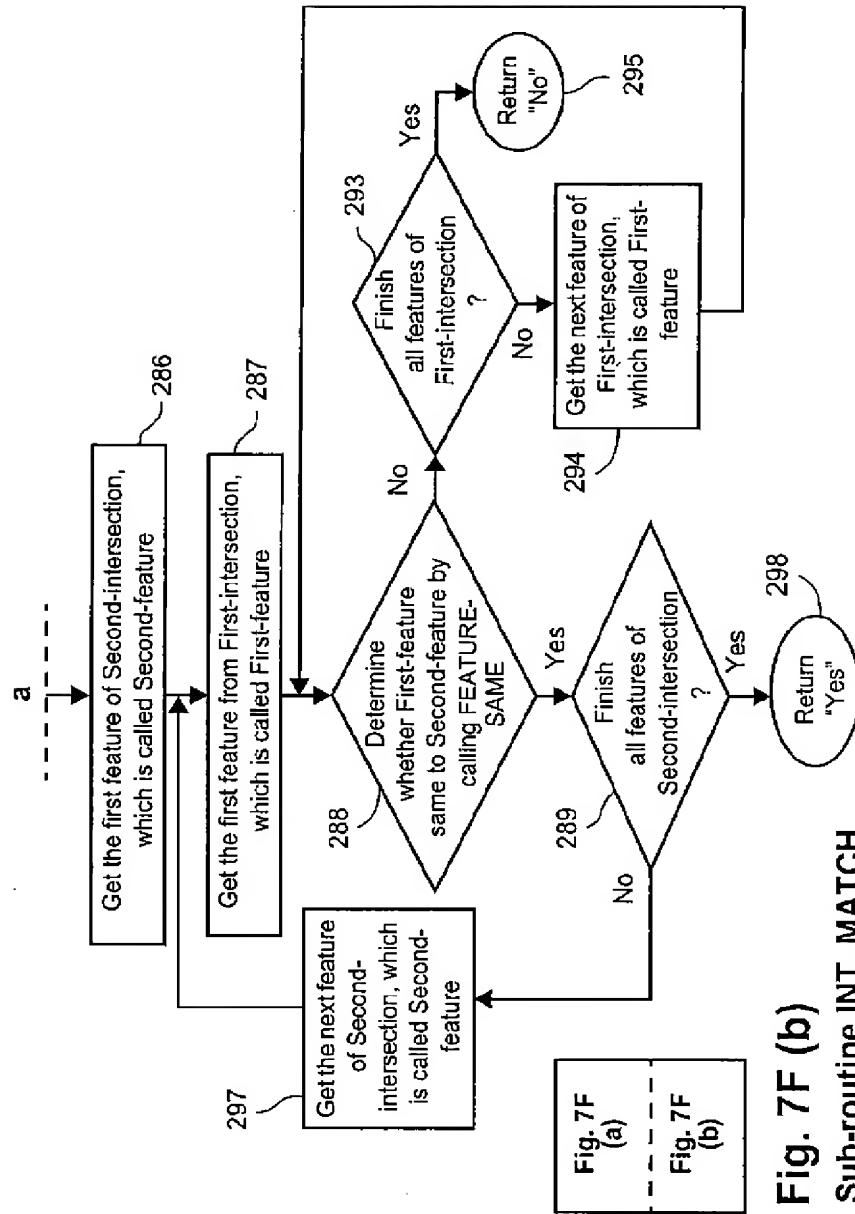


Fig. 7F (a)

Sub-routine INT_MATCH



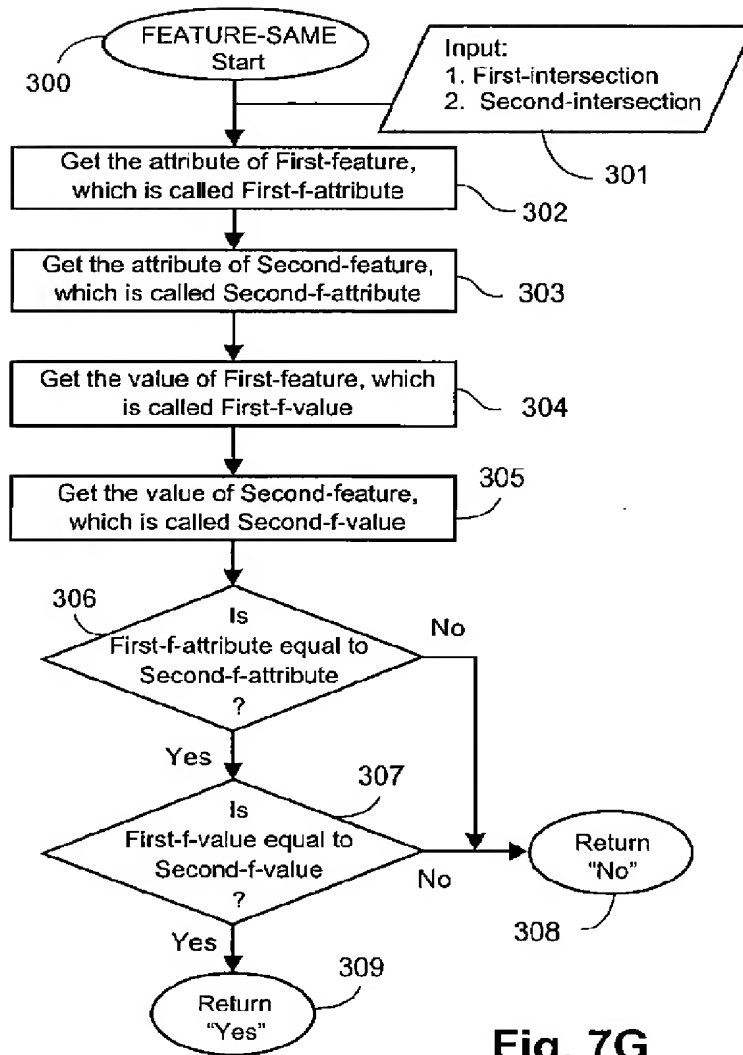
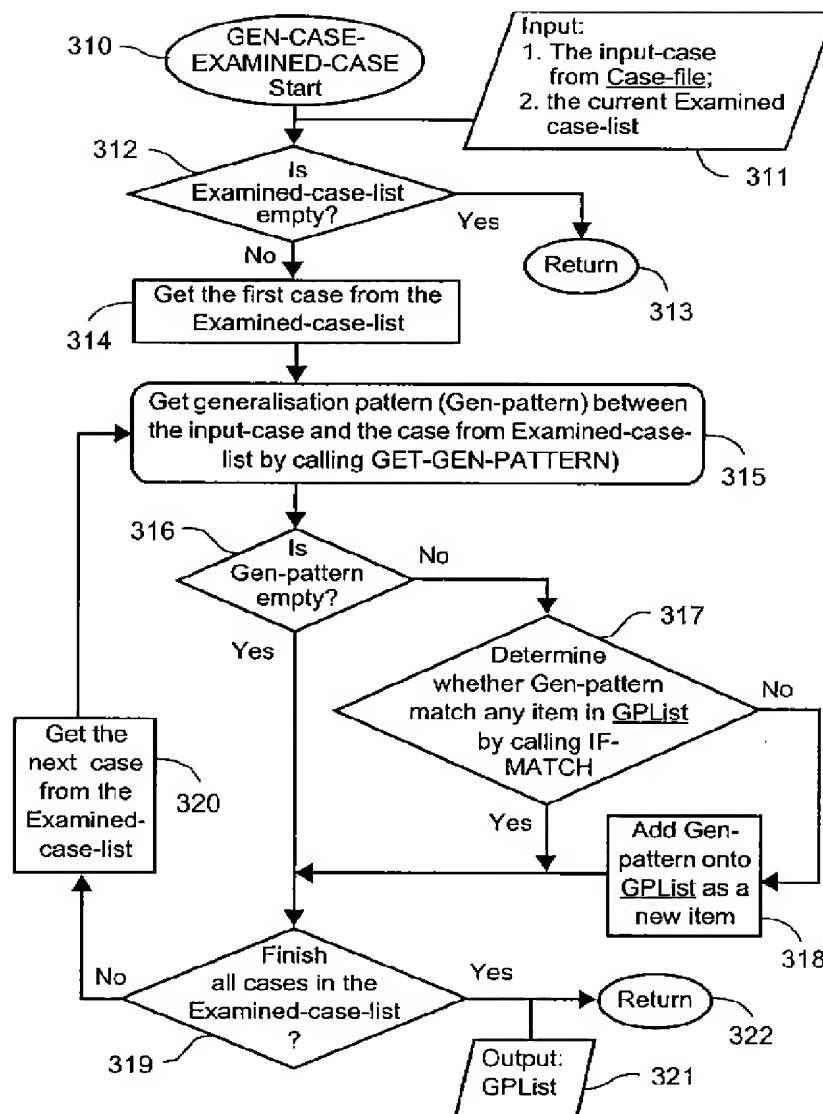
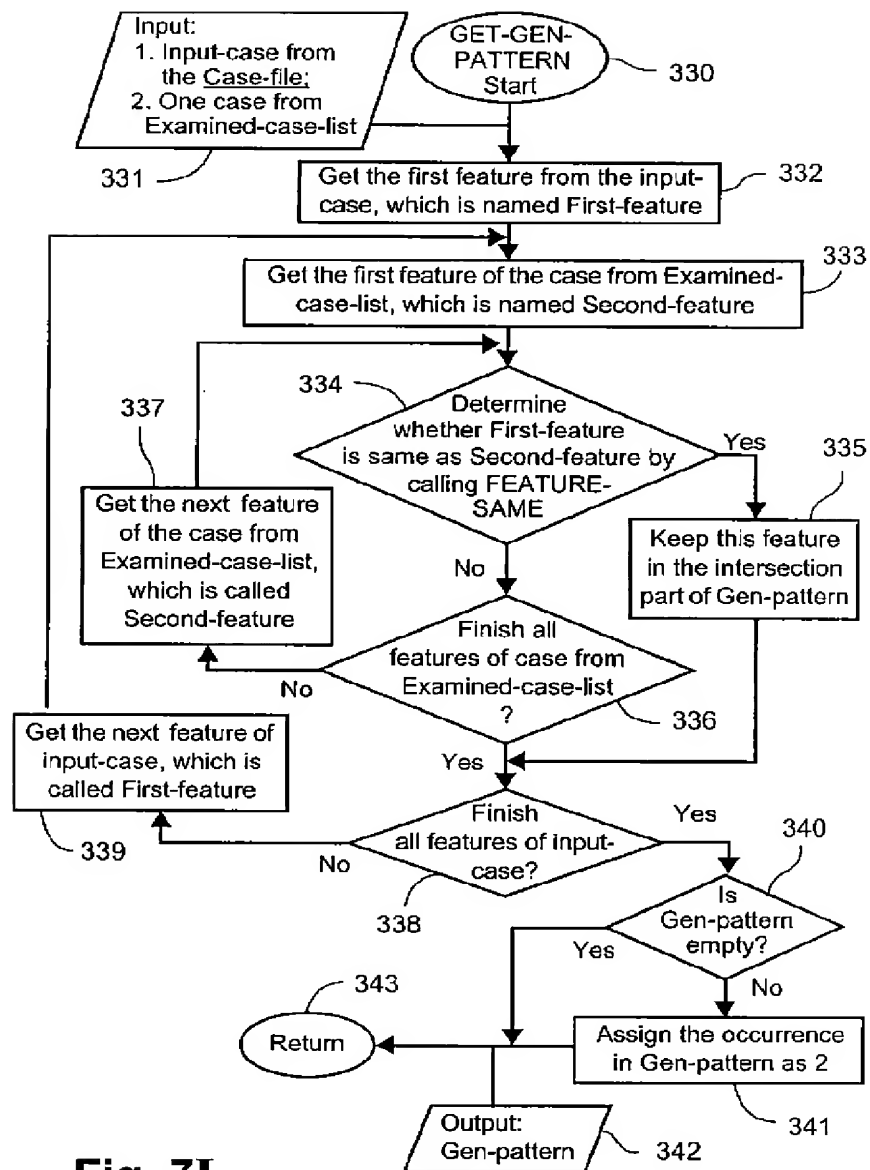


Fig. 7G
Sub-routine
FEATURE-SAME

**Fig. 7H****Sub-routine GEN-CASE-EXAMINED-CASES**

**Fig. 7I****Sub-routine GET-GEN-PATTERN**

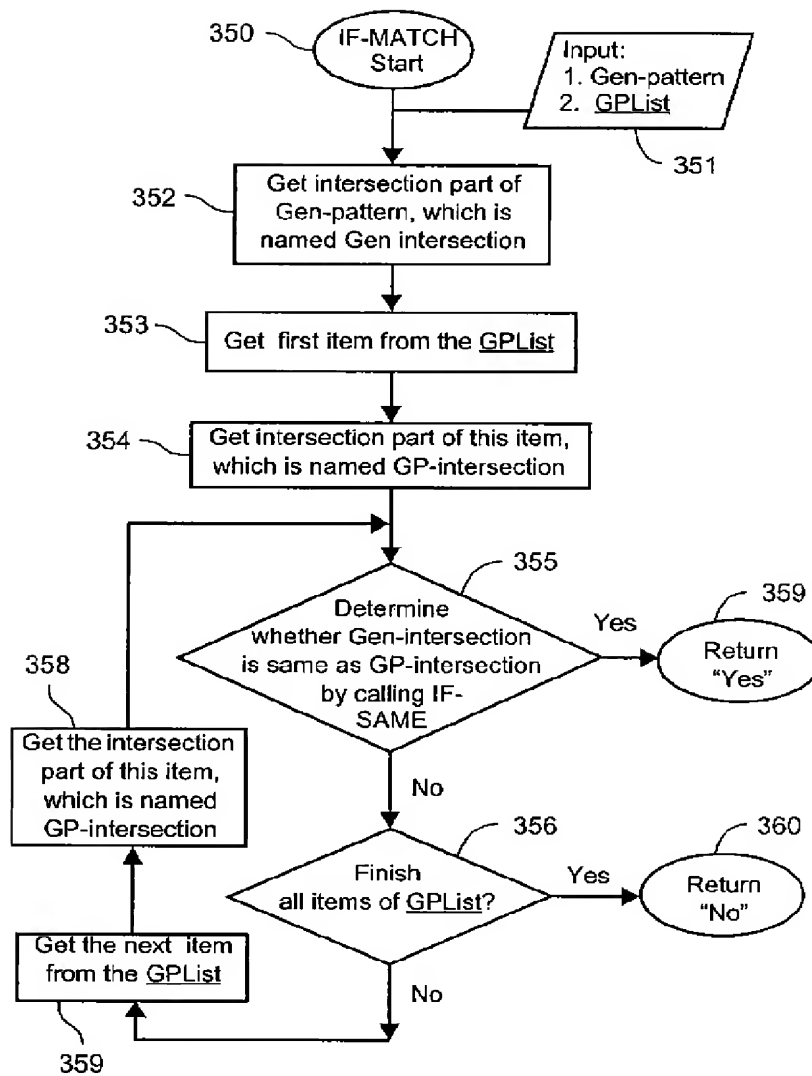


Fig. 7J
Sub-routine IF-MATCH

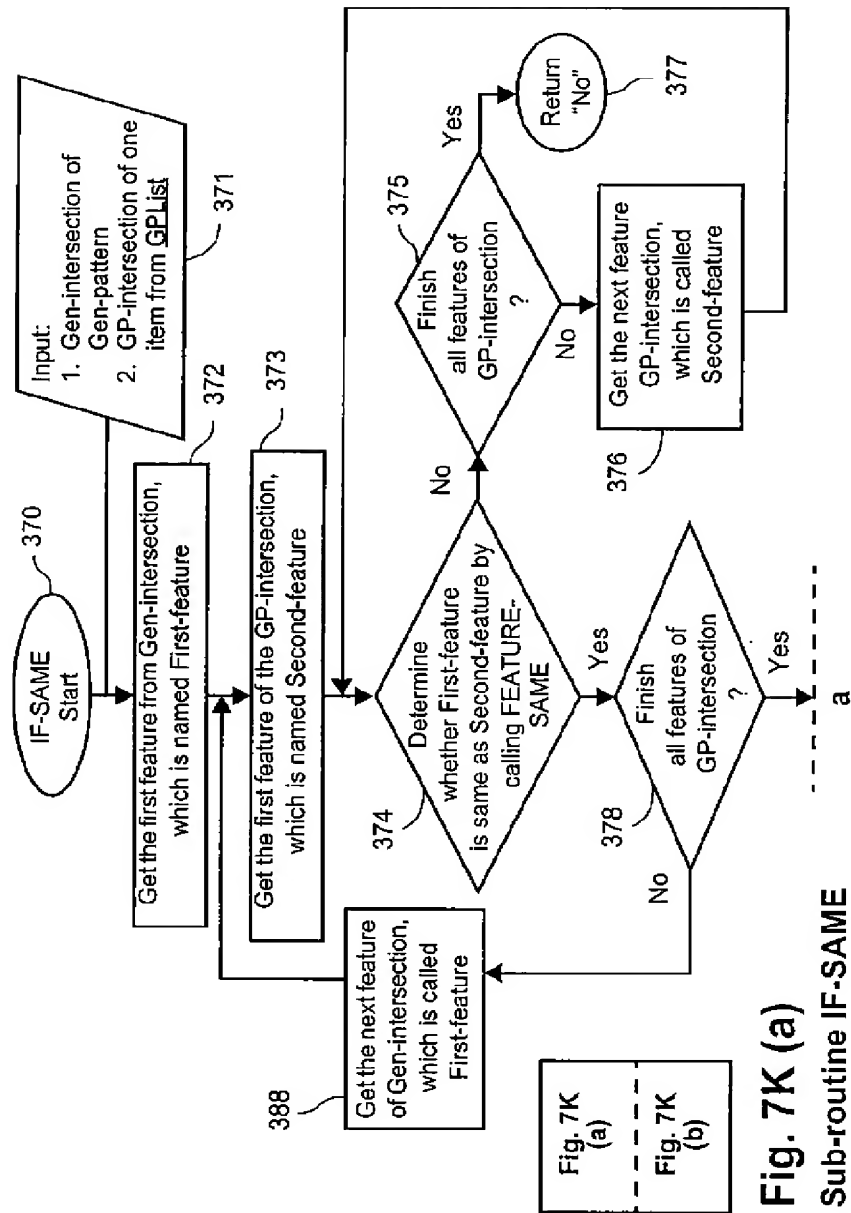
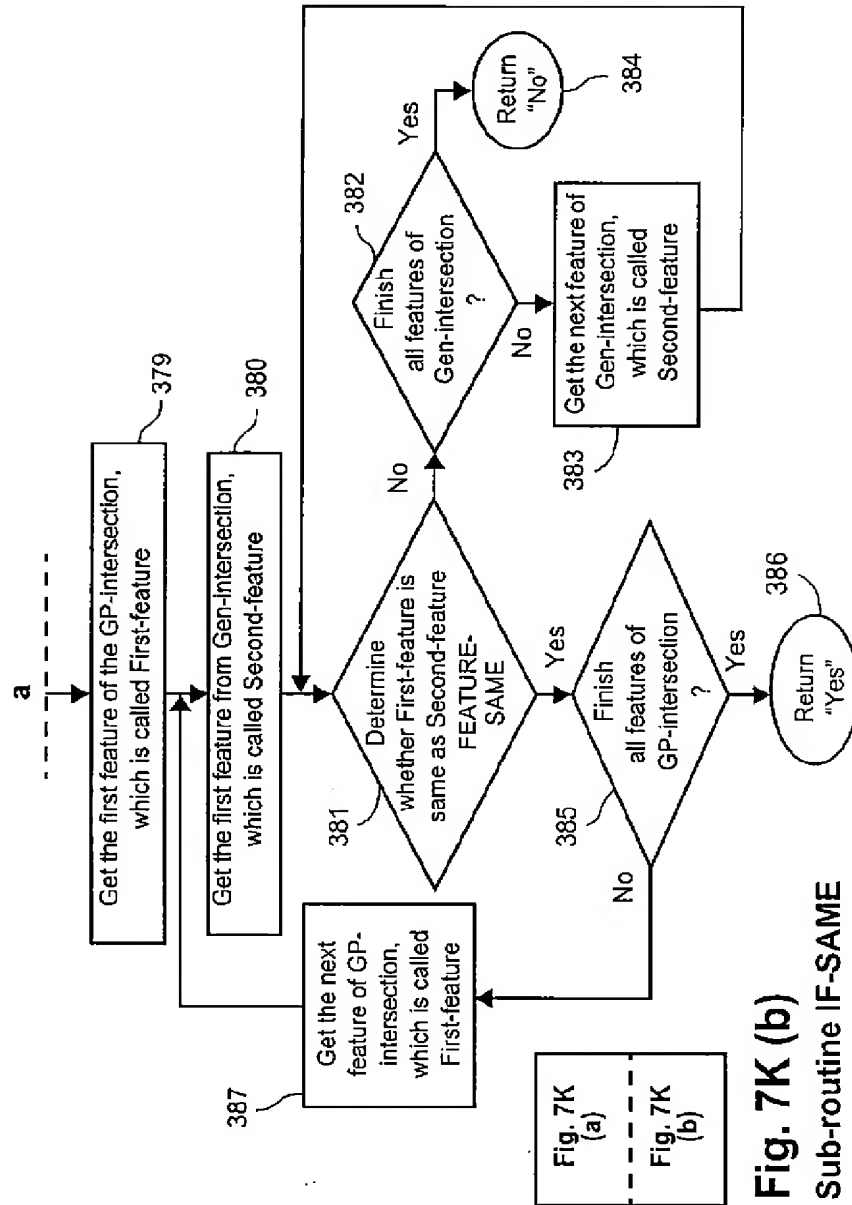


Fig. 7K (a)
Sub-routine IF-SAME



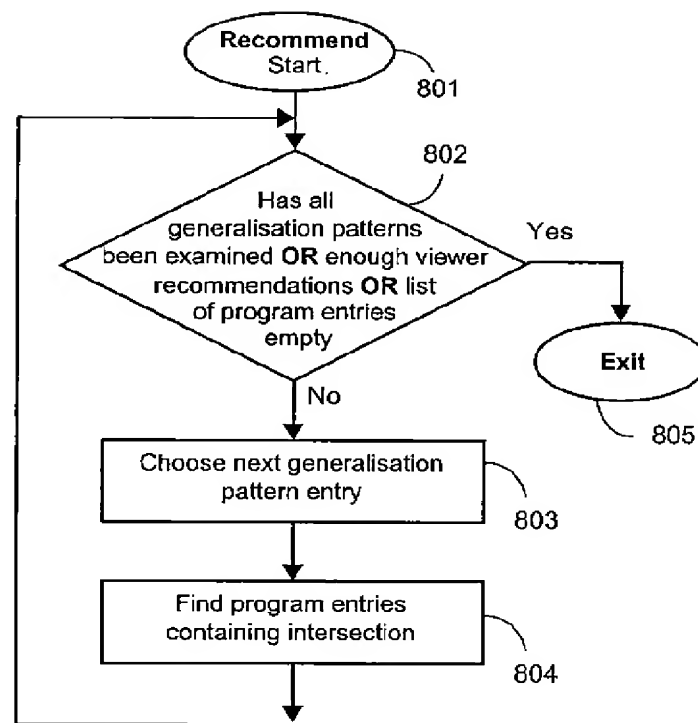


Fig. 8

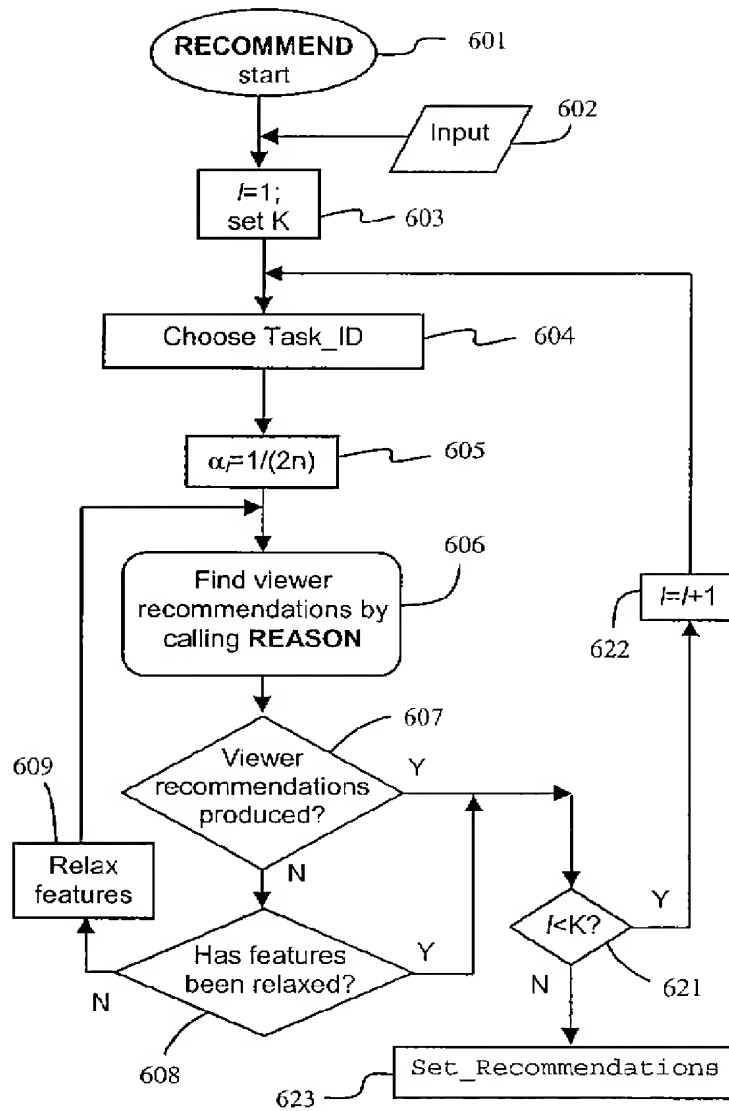


Fig. 9A
Procedure **RECOMMEND**

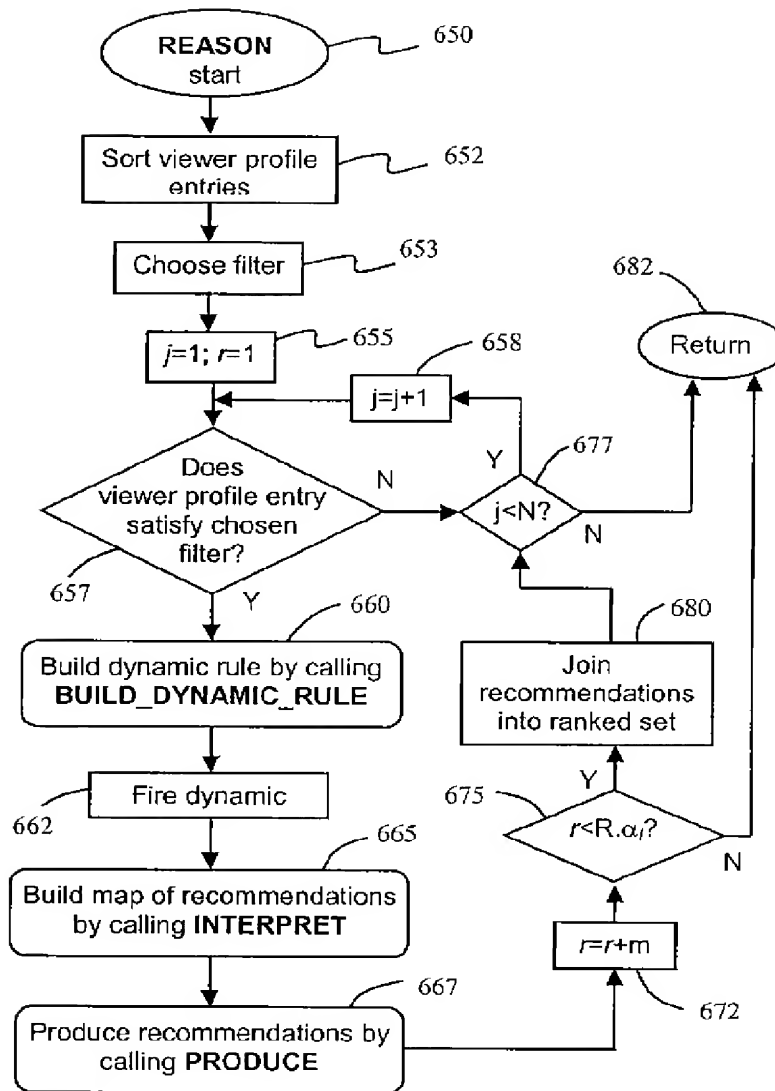


Fig. 9B
Sub-routine **REASON**

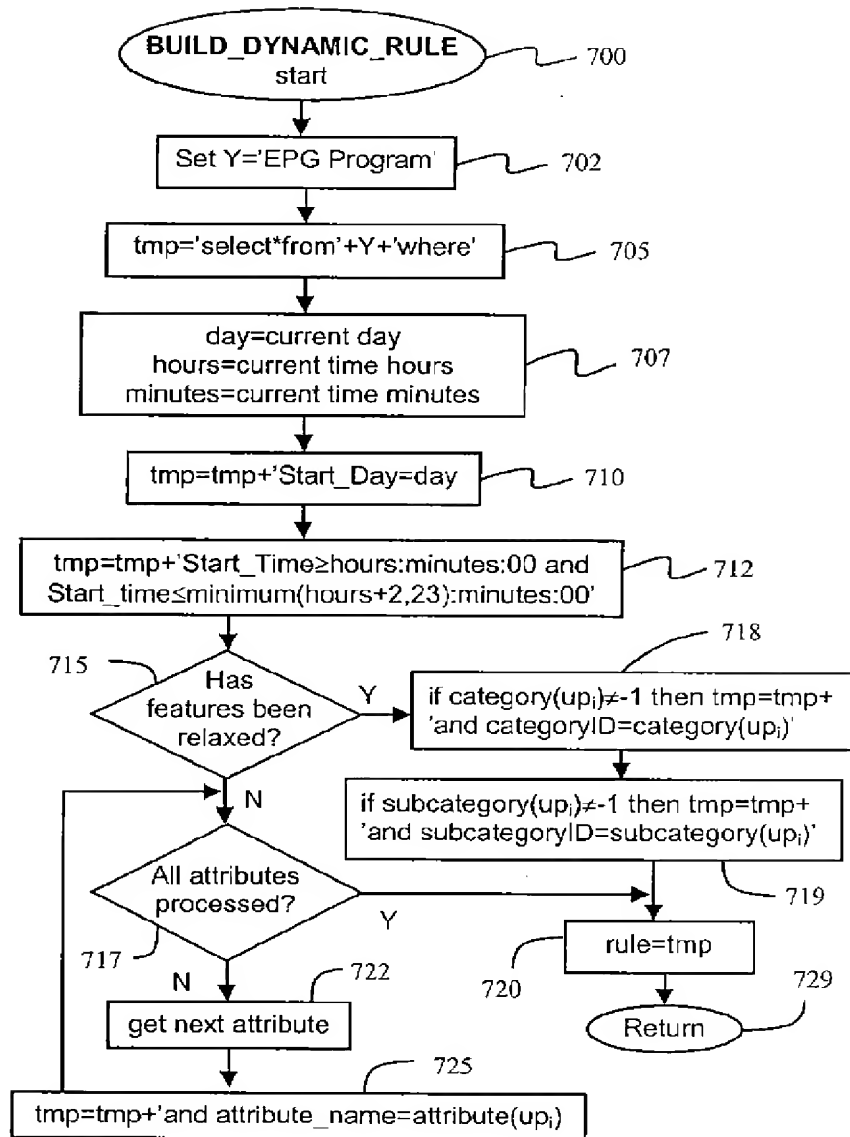


Fig. 9C
Sub-routine **BUILD_DYNAMIC_RULE**

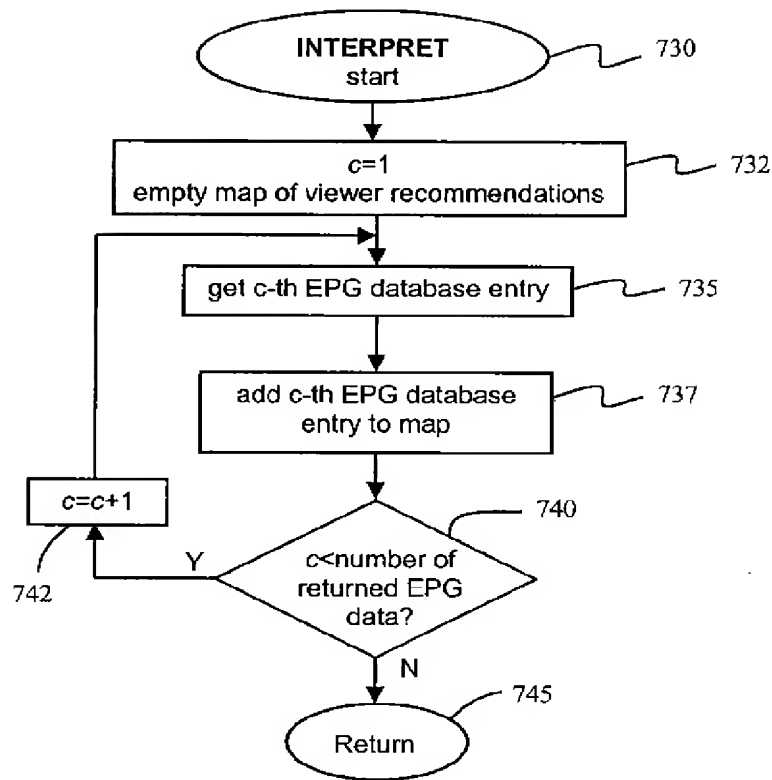


Fig. 9D
Sub-routine **INTERPRET**

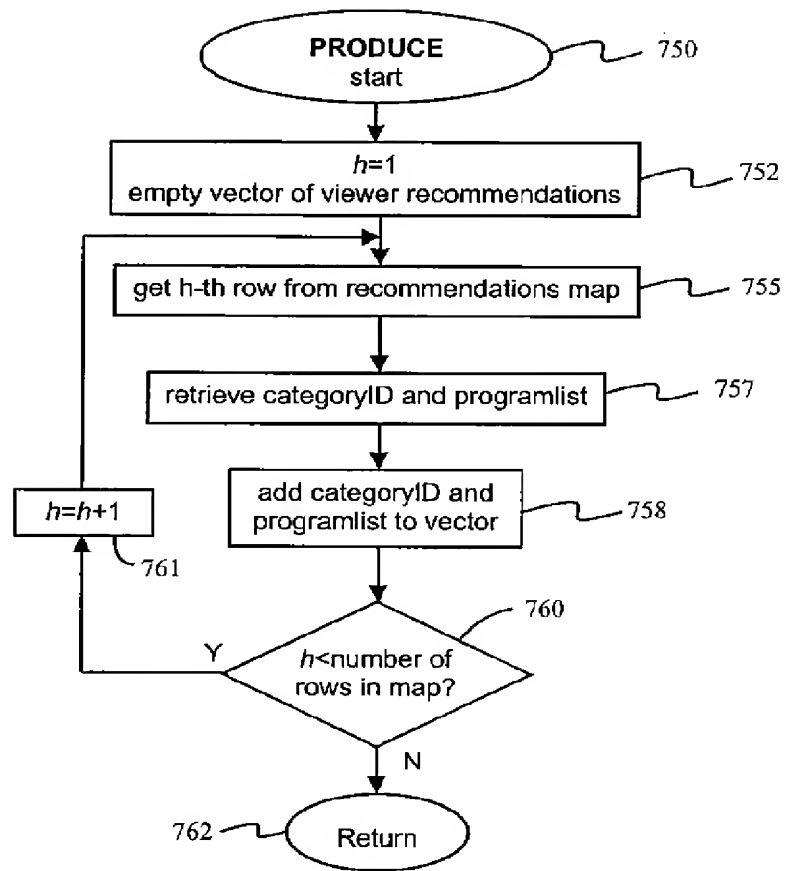


Fig. 9E
Sub-routine **PRODUCE**

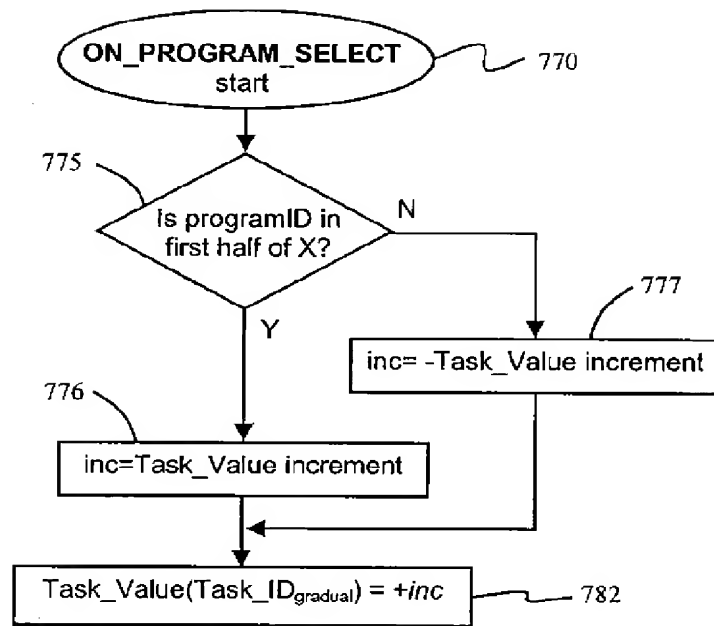


Fig. 9F
Routine **ON_PROGRAM_SELECT**

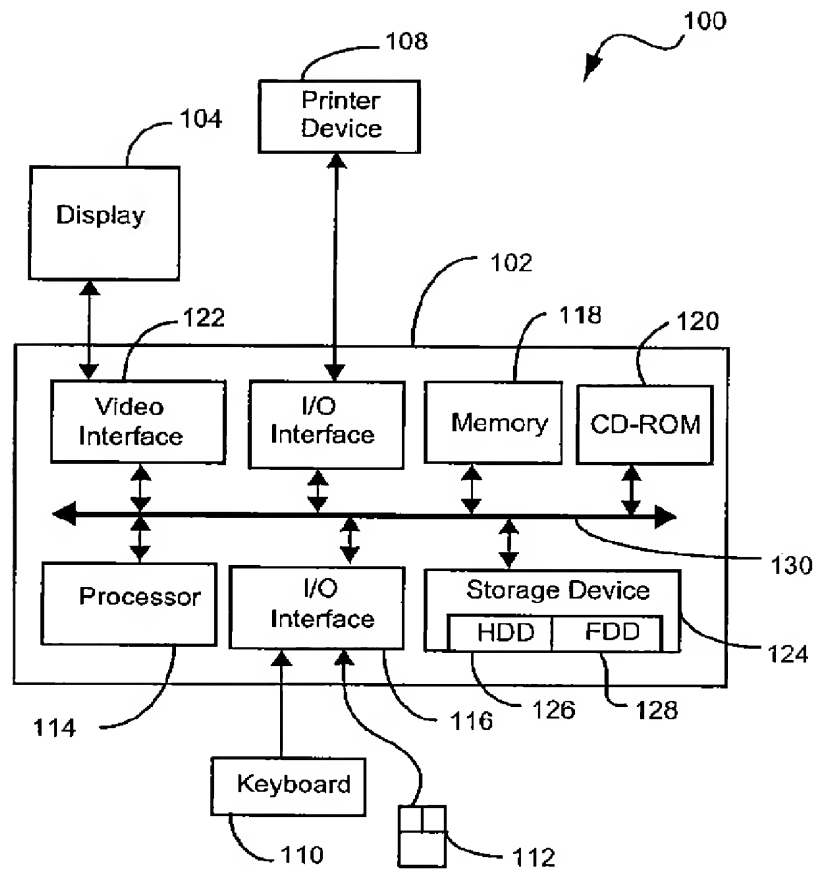


Fig. 10

Abstract

The present invention relates to a television system (50) and a method for automatically suggesting suitable programs to a viewer from a large number of available programs. The system (50) includes a DTV-agent system (21). Title information and characteristics of programs are made available as EPG (Electronic Program Guide) data, which including at least one Electronic Program Guide Database (22). A learning module (39) records characteristics associated with each program viewed by the user, and forms sets of these characteristics. The frequency of occurrence of each set is also determined. A recommendation module (40) uses a number of tasks to compile a list of viewer recommendations (67). Various tasks are defined, with each task defining a unique combination of a manner of ordering the viewer profile (500), and particular Relevance Filters for filtering the ordered viewer profile. Upon entry of a user request for a list of program recommendations (67), a search is performed of the EPG data for programs with characteristics that best match sets selected by the task(s). The user is notified of the availability such programs, allowing selection of a particular program.